# University of Cincinnati

**Date: 3/13/2015**

**I, Andrew W Schriner , hereby submit this original work as part of the requirements for the degree of Master of Science in Environmental Engineering.**

It is entitled:

**#CROWDWORK4DEV: ENGINEERING INCREASES IN CROWD LABOR DEMAND TO INCREASE THE EFFECTIVENESS OF CROWD WORK AS A POVERTY-REDUCTION TOOL**

Student's name: **Andrew W Schriner**

This work and its defense approved by:

**Committee chair: James Uber, Ph.D.**

**Committee member: Dominic Boccelli, Ph.D.**

**Committee member: Daniel Oerther, Ph.D.**

14013

# #CROWDWORK4DEV: ENGINEERING INCREASES IN CROWD LABOR DEMAND TO INCREASE THE EFFECTIVENESS OF CROWD WORK AS A POVERTY-REDUCTION TOOL

by

ANDREW SCHRINER

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF CINCINNATI

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ENVIRONMENTAL ENGINEERING

2015

Approved by

James Uber, Advisor

Daniel Oerther,

Dominic Boccelli

## Abstract

Globally, in 2010, 2.4 billion people still lived in extreme poverty, earning less than $2 per day. This thesis shows that crowd work has the potential to significantly reduce global extreme poverty – an approach I call CrowdWork4Dev. CrowdWork4Dev is fundamentally about taking a new approach to development interventions to fight poverty – taking the stance that engineering increases in employment is the most effective way to address deficits in basic needs. Crowd work is especially well-suited to fighting poverty for a number of reasons: 1) it is inherently distributable to any place with internet access 2) simple crowd tasks require no specialized skills, making them broadly accessible to almost anyone and 3) the price of simple crowd work makes it both economically viable for requesters and advantageous for developing-country workers.

In this thesis I analyze the interaction between the complex problem domain of poverty and the candidate solution of crowd work to determine where to apply solution effort. I argue that increasing the amount of crowd work available on crowdsourcing platforms that are accessible to developing-country workers is the lynchpin. In support of this goal, this thesis contributes the conceptual model of "the crowdsourcing stack" – common components of most crowd applications – and introduces Flowbuilder, a set of software tools implementing a portion of the crowdsourcing stack intended to make implementing a crowd-backed data project faster and easier. The capabilities of Flowbuilder are showcased through three example use-cases.

This thesis shows the potential of crowd work to empower many who currently struggle in the face of limited opportunity, and takes steps toward implementing CrowdWork4Dev as a solution.

# ACKNOWLEDGEMENTS

Jim Uber for being advisor, mentor, and therapist. Somehow he always knew what I was feeling even before I did. In a move that was later revealed to be rather prescient, he encouraged me to sink my teeth into programming, claiming (and I do quote) “computer programmers are the high priests of abstraction!”

Dan Oerther for being a mentor, friend, and advisor. There have been many teachable moments along the road and he rarely passed up the opportunity to impart some valuable lesson, whether practical, professional, or philosophical. I always felt that he viewed my success as his goal, and for that I am deeply appreciative.

Dominic Boccelli for many productive discussions, an excellent statistical methods class, and for tolerating my 50,000-iteration MCMC simulation built with MS Excel.

Hazel and George Okullo, Elishah Okullo, as well as Johnson, Philip, Sophy, Millicent, Judy, Elijah, and Fred for the assistance, participation, and many good times in Kenya.

My research collorators: Bruce Aronow, Anil Jegga, Mayur Sandraghar, Glenn Morrison, Sriram Chellappan, Raja Bolla, Chris Brown, David Hackney, Montana Puckett, Lee Voth-Gaeddert, Michael Orlando, Quentin Ortega, and Sarah Oerther.

My family: Mom, Dad, Beth, Doug and Scott for their support and patience.

And Zeke.

Lastly I would like to recognize the NSF Graduate Research Fellowship Program for providing me with the opportunity to undertake this research.

**Contents**

1. INTRODUCTION ........ 11

1.1. The Problem ........ 11

1.1.1. The problem of poverty ........ 11

1.1.2. Crowdsourcing employment as a solution (CrowdWork4Dev) ........ 12

1.1.3. Problems in crowdsourcing and human computation as it relates to poverty alleviation and economic development ........ 15

2. WHICH HAS GREATER LIQUIDITY: MONEY, EDUCATION OR DRINKING WATER? (Manuscript I, submitted to International Water Association) ........ 18

2.1. ABSTRACT ........ 18

2.2. INTRODUCTION ........ 19

2.3. ENGINEERS WITHOUT BORDERS WATER DISTRIBUTION PROJECT ........ 20

2.3.1. Iatrogenesis ........ 21

2.3.2. Water access is not sustainable development ........ 21

2.4. STRUCTURAL EQUATION MODELING OF THE DETERMINANTS OF THE IMPACT OF WATER FILTERS ON HEALTH ........ 22

2.5. CROWDSOURCING-BASED EMPLOYMENT AS AN ALTERNATIVE APPROACH ........ 23

2.6. CONCLUSIONS ........ 25

3. Crowdsourced Human Computation Fights Poverty And Enables Novel Data Processing (Manuscript II, submitted to Science) ........ 25

3.1. Abstract ........ 25

3.2. Introduction ........ 26

3.3. Feasibility testing ........ 27

3.4. Bioinformatics pilot project ........ 30

3.5. Impact Assessment ........ 31

3.6. Discussion ........ 33

3.7. Conclusions ........ 34

4. No really, (crowd) work is the silver bullet (Manuscript III, presented at Humanitarian Technology: Science, Systems and Global Impact 2014, HumTech2014) ........ 35

4.1. Abstract ........ 35

4.2. Introduction ........ 35

4.3. Why “crowd work” works........ 36

4.3.1. First – why work? ........ 36
4.3.2. What is “crowd work”?........ 37
4.3.3. Unique features of crowd work make it especially appropriate as a poverty-fighting tool 38
4.4. Results from our pilot project and income survey ........ 39

4.5. Conclusion........ 40

5. What Difference Does the Device Make? Crowd Work on Computers and Phones (Manuscript IV, submitted to HCOMP 2014) ........ 40
5.1. Abstract ........ 40

5.2. Introduction ........ 41

5.3. Methods........ 42

5.4. Results and Discussion........ 45

5.4.1. Do phone users tend to work in shorter bursts than computer users?........ 45
5.4.2. Do computer users tend to work around certain times of the day, while phone users work any time?........ 46
5.4.3. Are there differences in accuracy across the task types between user groups? ........ 47
5.4.4. Is there a difference in how long a user takes to complete a task between phone and computer users? ........ 48
5.4.5. Are there differences in worker satisfaction between the phone and computer groups?.... 49
5.5. Conclusions ........ 50

6. Flowbuilder, the crowdsourcing stack developer's toolkit (Manuscript V, submitted to Computer Supported Cooperative Work, CSCW 2014)........ 51
6.1. ABSTRACT ........ 51

6.2. INTRODUCTION........ 51

6.3. The crowdsourcing stack........ 53

6.4. Flowbuilder, the crowd stack toolkit........ 55

6.4.1. Design Philosophy ........ 55
*6.4.2.* Features ........ 56
6.5. Example use cases........ 61

6.5.1. Finding open windows in Streetview images ... 61
6.5.2. Responsive web design experiments ... 62
6.5.3. Extracting, mapping and labeling biochemical pathway diagrams from medical literature63
6.6. Limitations ... 64
6.7. Conclusions ... 64
7. Future work... 65
7.1. Impact Evaluation ... 65
7.2. Development of crowd stack toolkit ... 65
7.3. Validation of labor demand focus hypothesis ... 66
8. Discussion... 66
8.1. Suggestions for private companies... 66
8.2. Suggestions for traditional economic development actors such as NGOs and government aid bodies 67
9. Conclusion ... 68
References... 69

**List of Figures**

Figure 1 Results for survey question "How familiar are you with {crowdsourcing, human computation}?" .......... 16

Figure 2 Distribution of spending for 5 crowdsourcing workers in a Kenyan village. .......... 24

Figure 3 Example human computation project .......... 30

Figure 4 Distribution of spending across broad categories. The largest share of earnings went to education. The remainder went to meeting immediate basic needs, like food and clothing; making useful investments for the future, like small business expansion and livestock; or goods like pots and radios. .......... 32

Figure 5 Detailed breakdown of spending across eleven categories for five workers. .......... 33

Figure 6 Comparison of the size of foreign direct investment (FDI) and official development aid (ODA) over time. Beginning in the early 1990's FDI began to rapidly outpace ODA. .......... 37

Figure 7 Distribution of spending across 4 major spending categories formed by the grouping of 33 subcategories. The workers spent the majority on education, and spent nearly all of it on productive investments for the future. .......... 39

Figure 8 Streetview task on narrow smartphone screen. .......... 43

Figure 9 Streetview task on wide screen; no scrolling necessary. Image from "http://instantstreetview.com/s/10 E CONCORD AVE KANSAS CITY MO 64112" .......... 44

Figure 10 Session start times for phone and computer groups. All times are Central Time. .......... 47

Figure 11 Histograms of task duration calculated from time of page load to clicking "Submit" button. .......... 49

Figure 12 The crowdsourcing stack. The request-response cycle moves from the right side to the left and back to the right. .......... 54

Figure 13 Definition of one task type, in which crowd workers would view Google Streetview imagery and answer whether or not a house has any open windows. .......... 58

Figure 14 Generated worker interface for Streetview Task. Note that the second question, "How many windows are open?" is hidden unless the answer to the first question is "yes". This dynamic form behavior is accomplished by specifying a Python dictionary of conditions in the Task class. .......... 59

Figure 15 Streetview task on a narrow smartphone screen, as generated by Flowbuilder's responsive task template .......... 62

Figure 16 CytoscapeJS interface for pathway mapping task. On the left is the jpg figure from the original journal article; on the right is the redrawn (and now computable) pathway ............................... 63

## List of Tables

Table 1 Recommended level of investment for rural African villages by the United Nations Millenium Project ........ 11
Table 2 combinations of infrastructure access and amount of available work ........ 17
Table 3 Study participant demographics.  The participants were chosen to represent diverse occupations, ages and computer experience. ........ 29
Table 4 Accuracy of 3-response majority vote scheme on example image classification task. ........ 31
Table 5 Characteristics of inputs and outputs for tasks chosen for this experiment ........ 45
Table 6 Model coefficients and p-values for session length. Mean difference is computer group mean minus phone group mean. ........ 47
Table 7 Logistic regression results for accuracy as a function of phone or computer group and a random worker effect. ........ 48
Table 8 Median task durations (in seconds) for each task type and group. ........ 49

# 1. INTRODUCTION

## 1.1. The Problem

The motivating problem for this research is global extreme poverty. Recognizing that this problem is quite large in scope, it is not the objective of this thesis to "solve" this particular problem, but rather to make progress towards solving it by solving some of its sub-problems. The sub-problems addressed here must however be understood in relation to the broader problem of poverty.

In this section, first the problem of poverty will be discussed, then the use of crowdsourcing employment as a solution will be introduced, followed by a discussion of the problems associated with crowdsourcing employment as a solution.

### 1.1.1. The problem of poverty

The type of poverty of interest is extreme (absolute, not relative), persistent, structural poverty. Addressing the problem of poverty has been on the global conscience for approximately 70 years (since the end of WWII). Throughout those 70 years many different approaches have been tried. There continues to be vigorous debate about the merits of different approaches. This says, at least, that we haven't solved it yet, but not necessarily that we don't know how to solve it. Is there a commonly agreed upon "solution" that has for some reason (lack of funding, willpower, or other circumstance) not yet been implemented? Let us consider this question next.

Jeffrey Sachs argues that the solution involves large, targeted aid expenditure, in the form of the Millenium Villages Project (MVP). Table 1, reprinted from [1], shows the level of investment entailed in this program, representing $500,000 per year per village of 5,000 people of external aid. Because of its

| Interventions | U.S. $ per person per year | U.S. $ per year per village of 5,000 people |
|---|---|---|
| Household share: | 10 | 50,000 |
| Government share: | 30 | 150,000 |
| Donor share: | 70 | 350,000 |
| Total investment | 110 | 550,000 |
| Distribution by sector | | |
| Agriculture and nutrition (15%) | 17 | 82,500 |
| Health (30%) | 33 | 165,000 |
| Infrastructure (energy, transport, communications) (20%) | 22 | 110,000 |
| Education (20%) | 22 | 110,000 |
| Water, sanitation, environment (15%) | 17 | 82,500 |

The MVP contributes $50 of the $70 donor share.

**Table 1 Recommended level of investment for rural African villages by the United Nations Millenium Project**

large reliance on external aid the scalability of the MVP approach is questionable, and the actual provision of such aid has fallen short of 'required' levels [2]. Further, the top-down 'big-push' characteristics of the approach are not unlike development approaches of the past that have failed to solve the problem [3][4][5]. Even the ostensible successes of the MVP project are uncertain, as claims regarding reduced child mortality in project villages have been exaggerated, and raw data has not been made public [6].

Another large scale development player, The Bill and Melinda Gates Foundation, made grants of total value of at least 8.95 billion from 1998 to 2007, focusing mostly on fostering technology-based progress in global health. Public data on the direct impact of these investments remains scarce; [7] argues that the focus on technology development misses the mark, when current technologies are sufficient but the wider social, health and economic systems of implementation, dissemination and provision of care are lacking.

The work of MIT economist Esther Duflo consists largely of applying randomized controlled trials to the assessment of economic development interventions. Such studies are the "gold standard" in epidemiological evidence for causation: they give the clearest indication of whether or not an exposure, or treatment, is causally linked to an outcome, because assignment to 'treatment' or 'no treatment' is randomized. Such RCTs have been applied to the assessment of microfinance [8], improved cookstoves [9], inclusion of females in village leadership [10], immunization campaigns [11], HIV education[12]. The outcome of all of this research using RCT's to evaluate development policies and interventions is a set of results of the form "A is better than B" for very specific contexts. We don't yet, however, have a mechanistic understanding that unifies all of these individual results into a comprehensive predictive model for "how to solve poverty" in an arbitrary context.

In short, there is ongoing vigorous debate about the way to fight extreme poverty, with a wide variety of views represented. Thus, in summary, the problem remains, not because we have the solution in hand and have not yet been able to implement it, but because we do not yet know what the solution is.

#### 1.1.2. Crowdsourcing employment as a solution (CrowdWork4Dev)

Fundamentally, poverty exists as deficits in many different basic needs (e.g. food, clean water, health care, etc.) across individuals, locations, and time. Some development interventions, such as building a water treatment system, provide for one basic need at a time; on the other hand, increasing employment provides money, which is a liquid resource that can be directed toward meeting any basic need in which an individual is currently experiencing a deficit. CrowdWork4Dev is fundamentally about taking a new approach to development interventions to fight poverty – I argue that engineering increases in employment is the most effective way to address basic needs.

#### 1.1.2.1. Introduction to crowdsourcing and human computation

Crowdsourcing, as defined by Jeff Howe, "represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call" [13]. Crowdsourcing may include many types of activities, including data processing, creative production (e.g. idea contests), crowdfunding (e.g. KickStarter), and task completion in the physical world. This thesis is primarily concerned with the subset of crowdsourcing that pertains to sourcing knowledge work. Crowdsourcing knowledge work (also sometimes called "crowd work") encompasses both "human computation" or "microtasking" – work in which the tasks are very small, short, and simple – and more complex and highly skilled work such as graphic design, computer programming, and technical research and development [14][15]. Small, simple tasks and complex, expert tasks can be thought of as opposite ends of a spectrum, and crowd work can fit anywhere along this spectrum, ideally with a match between the demands of the task and the skills of the crowd worker.

Residing at the easy end of the spectrum, "human computation" is a computing paradigm in which humans act as processors, completing tasks that are difficult for computers but easy for humans [16]. As described in the introduction to his dissertation, von Ahn offers a compelling definition of and argument for the power of human computation:

> *Construction of the Empire State Building: 7 million human-hours. The Panama Canal: 20 million human-hours. Estimated number of human-hours spent playing solitaire around the world in one year: billions. A problem with today's computer society? No, an opportunity. What if this time and energy could be channeled into useful work? ... we treat human brains as processors in a distributed system, each performing a small part of a massive computation.*

Human computation is useful for tasks which are very numerous but require some amount of human judgment. Human computation in various forms has been used to transcribe text [17], label images [18], and predict protein conformations [19], provide captions for audio as a commercial service [20], count neurons in microscope slides [21], and more.

Furthermore, Little et al. [22] have demonstrated that human computation operations can be efficiently integrated into hybrid computer-human processing algorithms in an imperative programming paradigm, in which algorithm steps are sent to the most appropriate resource for completion. Bernstein et al. [23] has also shown the usefulness of multi-step human computation workflows like Find-Fix-Verify, in which the first human computation task is to find an error in a block of text, the second task (completed by a different human) is to fix the specified error, and the third task is to verify that the fix is correct. In this way the strengths of human analysis can be combined with the strengths of computer analysis to solve problems that neither could solve alone.

Human computation can also be considered as a sort of knowledge-work assembly line, much as the physical assembly line made it possible to substitute a group of minimally trained workers completing a purposefully arranged series of operations for a highly trained (and rare) craftsman. As a result, the quantity of output of the factory was dramatically increased. In a similar way, human computation allows simple process steps in the knowledge work assembly line to be distributed and performed by a global workforce. A key feature of this arrangement is that the individual tasks are simple enough that they can be completed by nearly anyone, with only a minimal level of required training.

Human computation can also be combined with more complex types of crowd work, with expert-level workers completing tasks with high domain-specific knowledge demands (e.g. interpreting medical terminology, or writing algorithms in pseudo-code), or managing the crowd by planning workflows, training workers, or reviewing work. Given the human abilities and skills change over time, workers can also move from simpler to more complex types of crowd work as they gain experience and skills.

#### 1.1.2.2. A candidate solution

Crowd work is especially well-suited to fighting poverty for a number of reasons.

- It is inherently distributable to any place with internet access (and while many areas of the world do not yet have access, large firms such as Google are making targeted investments to increase connectivity in developing countries [24]).
- At the human computation or microtasking end of the spectrum of task complexity, the work requires no specialized skills and minimal training, making it broadly accessible to almost anyone. Further, as skills are developed, crowd work provides opportunities for more engaging

and demanding work (similar to the development of the information technologies services industry in India).

- On existing crowd work platforms such as Amazon's Mechanical Turk, microtasks are often priced at a few pennies each, allowing workers to earn on the order of several dollars per hour. While many US-based "Turkers" complain about the low rates, for someone who otherwise would make $1-$2 per day for difficult manual labor, these wages are attractive.

### 1.1.3. Problems in crowdsourcing and human computation as it relates to poverty alleviation and economic development

Crowd work as an economic transaction requires both a buyer and seller of the work – labor demand and labor supply. The labor demand is determined by the quantity of work being requested via crowdsourcing marketplaces, and the labor supply is determined by the quantity of workers seeking work in the marketplaces. For crowd work to have a broad impact on reducing poverty, both the labor supply and labor demand must be increased.

#### 1.1.3.1. What are the barriers to increasing labor demand?

There are two main barriers to increasing labor demand: awareness of crowdsourcing or human computation as a data processing tool, and the ability to successfully execute a crowdsourcing data processing project.

To gather further information about awareness, in [25] I conducted a survey to determine to what extent people use crowdsourcing and/or human computation in their work. The survey was distributed by first-degree connections within our own social networks to second-degree or greater connections. A total of 98 surveys were received. The results were filtered based on self-reports of how much data analysis was conducted by the respondent. Only the results from respondents who "regularly" or "frequently" conduct data analysis were retained, for a total of 77 surveys retained. The results show that very few people performing data analysis have used crowdsourcing, or even know much about it. Even fewer respondents have heard about or used human computation. See Figure 1 for a graph of the responses.

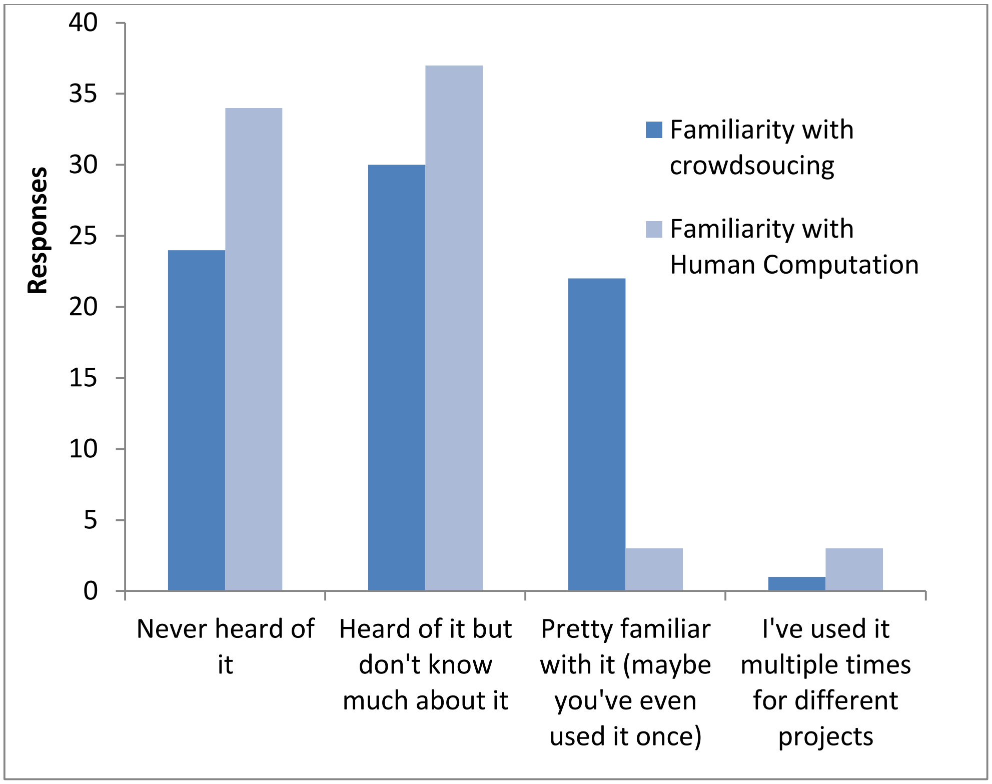



**Figure 1 Results for survey question "How familiar are you with {crowdsourcing, human computation}?"**

The second problem is that building a system to collect and process responses from the crowd requires a large investment of software development time, as well as understanding of a wide variety of topics and best practices such as incentive design, task UI design, training systems, how to write good instructions, task decomposition, and more. As one panelist at [26] noted, "Not everyone wants to get a graduate degree in crowdsourcing just to get results from the crowd."

Opportunities to increase labor demand therefore include increasing awareness of the existence and utility of these tools for data processing, and lowering the barrier to entry for creating successful crowdsourcing applications.

#### 1.1.3.2. What are the barriers to increasing labor supply?

For clarity, in this context labor supply refers particularly the supply of workers who are experiencing poverty (the labor supply in the global crowdsourcing market is of course composed of people of many different levels of socio-economic status).

Awareness must be mentioned as a barrier, because if people are not aware of the opportunity to participate in the market then they won't participate. It stands to reason, however, that if participation in the market is sufficiently attractive, the awareness problem will be solved by way of word of mouth. A more significant barrier than awareness is access to the necessary computing resources, electricity, and internet connection to participate in a digital networked crowdsourcing marketplace. While access to these resources presents a challenge in many (especially rural) areas, it is not insurmountable. MobileWorks [27], mClerk [28], and TxtEagle [29] are all examples of crowdsourcing systems built to leverage mobile phones (basic phones or smartphones). Other options include cybercafés, microfinancing, cooperatives, or other arrangements to secure access to computers. Additionally, firms like Google are currently making large investments in internet access for the developing world [24].

#### 1.1.3.3. What are the key problems? Where should we start?

In order for this approach to impact a significant number of people in developing countries, both labor supply and demand must be increased. Ability to access platforms can be addressed in a number of ways as discussed above, but *access to platforms is only valuable if there is work to do on the platforms*, i.e. the expected return on investment for workers must be high enough for them to make the necessary investments. Thus the single most important factor determining whether this approach will be successful is *how much work is available* to disadvantaged workers. Table 2 considers the possible combinations of infrastructure access and amount of available work.

**Table 2 combinations of infrastructure access and amount of available work**

| | **No infrastructure access** | **Infrastructure access** |
|---|---|---|
| **Little work** | Status quo | One Laptop Per Child; resources but no (financially) productive way to use them |
| **Lots of work** | *Potential workers have incentive to invest in access* | Labor supply and demand well matched, productive employment for workers |

Increasing the overall demand for human computation services (by awareness and education of potential users of human computation) is one option. The utility of human computation for data analysis and problem solving has been thoroughly demonstrated [30], though it is still not widely used either in scientific research or commercial settings [31][25]. I and others [26] believe this is due to an awareness gap and to obstacles in specification and setup of a human computation project. Additionally many current data analysts (we use this very broad term intentionally) are not accustomed to seeking out solutions that involve large-scale, brute force human effort. Hence the crowdsourcing industry is still a "*high growth, early stage industry*" [32].

The remainder of this thesis consists of submitted manuscripts which address the questions of 1) why crowd work is an appropriate tool in the fight against poverty and 2) how to solve the problems that currently stand as obstacles to wider adoption of crowd work as a solution. The manuscripts are not arranged in chronological order but are instead arranged to coherently address 1) and 2) above.

# 2. WHICH HAS GREATER LIQUIDITY: MONEY, EDUCATION OR DRINKING WATER? (Manuscript I, submitted to International Water Association)

D.B. Oerther, A.W. Schriner

## 2.1. ABSTRACT

*International development efforts often focus upon meeting basic needs (e.g. access to water and sanitation) such that a community can invest more time in economic pursuits. In this paper we trace the evolution of our approach to meeting basic needs over the course of eight years and three case studies. We describe our shift from a traditional approach to providing water access according to the Engineers Without Borders model, to an alternative approach where we consider income and education as primary causes of improvements in health rather than secondary effects of improved clean water access. Ultimately we conclude that the most effective way to promote sustainable development is by increasing income via access to employment using crowdsourcing, which will lead to improved village health and reduced diarrhea burden.*

## 2.2. INTRODUCTION

Some in this research community are working on the problem of providing water treatment technology to communities in developing countries as a way to alleviate extreme poverty. The idea is that extreme poverty is characterized by deficits in basic needs, and directly eliminating those deficits (e.g. by the construction of a water treatment and distribution system) will lead to "development," and the elimination or reduction of extreme poverty. Here "development" is typically assumed to mean "economic development," and the narrative is that when basic needs are met, poor people are able to devote more time to economic activities and then, finally, begin on the path of economic growth. Specifically in the case of water infrastructure interventions, this story typically goes, "We made clean water more easily accessible, and so the women and girls in the village can spend less time collecting water and more time in productive pursuits or going to school," or "With clean water, people get sick less often (i.e. reduced diarrheal health burden), and then they can be more productive." The assumed direction of causality, or the propagation of consequences, is from "meeting basic needs" to "increased productivity" to "economic development". The authors at one time subscribed to this basic view of environmental engineering in support of poverty alleviation.

In their book Poor Economics [33], Banerjee and Duflo make a strong case for the necessity of measuring the effectiveness of interventions on a case-by-case basis. Meanwhile in The White Man's Burden [34], Easterly shows that traditional aid interventions create damaging individual incentives (e.g. dependency on handouts) and argues for a re-examination of development work through the lens of incentivized behavior.

In light of such work from the development economics literature, it is worthwhile to revisit the narrative of development that underlies environmental engineering work on poverty alleviation. In this paper we relate the evolution of our views on the usefulness of water projects for international development work by considering three case studies from our work over the past 8 years. We consider 1) an Engineers Without Borders (EWB) water project, 2) a research study to quantify the determinants of the impact of household slow-sand filters on diarrheal health burden, and 3) a pilot project to use crowdsourcing to provide employment to rural villagers. For each project we provide a brief background and description, and then discuss the lessons learned and how the project has affected our views on how we should go about our development-related work.

### 2.3. ENGINEERS WITHOUT BORDERS WATER DISTRIBUTION PROJECT

In 2007 our EWB chapter began work on a water distribution system in Kamuga, Kenya, a community of approximately 500 people in a region where 64% of the inhabitants live below the $2 per day poverty line [35]. A team of undergraduate engineering students performed the community assessment (2007), design, implementation (2009), and post-implementation evaluation (2010) of 1) a solar-powered groundwater pump, 2) two aboveground concrete storage tanks, and 3) five kilometres of distribution system for domestic use, animal husbandry, and kitchen gardens. Total project cost was approximately $50,000. From our experience with EWB-USA, this project was typical of water infrastructure projects undertaken by EWB chapters.

While we conducted pre- and post-implementation evaluations in the community to measure the impact of the project, this type of data, even when properly collected, does not provide credible evidence about the effects of the intervention [33]. Credible evidence would require a valid counterfactual (or "control") community, with data collected in both the intervention community and the counterfactual community before and after implementation. However, due to the turnover of undergraduate students in EWB chapters, maintaining a long-term commitment and the expertise to do effective evaluation is prohibitively challenging. With apologies to the donors who contributed to this particular project, our fundraising claims about the "long-term sustainable development" initiated by that intervention remain unsubstantiated. Again, we find this lack of rigorous evaluation to be typical of not only EWB projects, but many of the projects undertaken by the environmental engineering research community (quite understandably, because rigorous evaluation is indeed quite challenging). To their credit, in 2013 EWB released new guidelines to improve project evaluation [36], but they still do not address the fundamental problem with credibility described above. Some evaluation is definitely better than none, but we must avoid claiming that we know more than we actually do about the impacts of our work.

Secondly, through the course of that project and our continued involvement with that community, it came to our attention that the community leaders were soliciting aid from three different international organizations and one domestic organization simultaneously for water and building construction projects, without disclosing these arrangements to any of the four organizations. Imagine the surprise upon crossing paths with a volunteer from another organization during a visit to the community. While we found these revelations to be somewhat concerning, such conduct is unsurprising, as it is an economically rational response to the incentives created by this system of aid. If significant investments in capital infrastructure are on offer from various organizations, then “playing the field” is an effective strategy for maximizing the expected benefit to one’s community.

Occasionally (probably not often enough) development practitioners express some concern related to their awareness of the cultural changes brought about by their interventions (e.g. the elimination of the communal well as social space). We certainly share this concern and do not have a resolution to offer to ameliorate it; we think it does not go far enough. We discovered (after construction) that some of the wealthier community members had been (prior to construction) paying some of the poorer community members to collect water for them. With the introduction of easily accessible water for all, we had unintentionally eliminated some sources of employment and possibly even contributed to increasing inequality within the community. These are challenging issues that should be taken into consideration during the design of such projects.

### 2.3.1. Iatrogenesis

Here we would like to promote the concept of iatrogenesis in the discourse on development interventions. Iatrogenic is defined by Merriam-Webster dictionary as “induced inadvertently by a physician or surgeon or by medical treatment or diagnostic procedures”. In Antifragile [37] Taleb notes that it has only been within approximately the past century that physicians began to do more good than harm for their patients, as a result of such practices as bloodletting and the absence of good hand-washing hygiene (despite the axiom of primum non nocere, or “first, do no harm”). Taleb goes on to broaden the concept of iatrogenesis to a variety of domains including finance and education, and warning of “naïve interventionism,” or interventions without consideration for possible iatrogenics.

The concern about changing cultural systems, the elimination of the water-hauling jobs in Kamuga, and Easterly’s concerns about the incentives created by aid all represent cases of iatrogenesis in the realm of development work. We encourage engineers who participate in development work to give serious consideration to the possibility of inadvertent harm, and adjust their activities accordingly.

### 2.3.2. Water access is not sustainable development

There are many competing definitions of sustainable development, but common among them are the concepts of 1) continuation for a long time rather than short time and 2) state change, from the present state to a future state. Providing water access is often held up as an example of “sustainable development” – we have spoken of this ourselves – but does it meet these two requirements? Infrastructure costs money to maintain over time; if a community cannot pay for maintenance over the long term, then the project is clearly not sustainable in an economic sense. If they cannot already pay for such costs, then economic state change is required. This ultimately requires livelihoods; and if they are not already present or sufficient, providing a water infrastructure project will not necessarily change that. There is at best an indirect and uncertain link from water access projects to long-lasting state change; this should be obvious from the many failed projects of the past that now sit broken and unmaintained.

We caution that usage of the term "sustainable development" with regard to water projects may function as a "semantic stop sign" [38] – that is, a label which, once applied, prevents the asking of additional challenging questions. Of course "sustainable development" is a good thing; but how exactly will an expensive water infrastructure project that a community could not afford on its own be "sustained", and exactly what "development" will result from easier access to water, and on what evidence do we base our answers?

### 2.4. STRUCTURAL EQUATION MODELING OF THE DETERMINANTS OF THE IMPACT OF WATER FILTERS ON HEALTH

In 2009, Divelbiss began work to use structural equation modelling (SEM) to investigate the complex set of factors mediating the impact of water filter interventions on diarrheal health burden (DHB). SEM has two important strengths for this type of analysis: 1) the use of latent (unobserved) variables like socioeconomic status (SES), which are estimated based on multiple observed variables such as ownership of a vehicle and dwelling construction materials, and 2) the ability to represent both direct and indirect relationships between variables. Household surveys were conducted in rural Guatemala (n=286) in homes where the Centre for Affordable Water and Sanitation Technology (CAWST) filter was in use.

The results of this work (see Divelbiss, Boccelli, Succop, & Oerther, 2013 for details) show that while a properly operating and maintained CAWST filter is associated with a decrease in DHB, increased household education is associated with an even larger decrease in DHB. Further, the full SEM model, which accounts for household education, socioeconomic status, hygiene practices, quality of water source, and extent of additional water treatment in the home, explained only 7% of the variance in DHB. Divelbiss notes:

"*The community is a complex system of interactions which directly and indirectly influence the* *health of its residents. Policy makers and development practitioners must recognize that single target interventions (e.g., improving water quality) have a limited influence on the entire system."*

In addition, Voth-Gaeddert has extended this approach, using data from the US Agency for International Development (USAID) Demographic and Health Surveys (DHS) Program and a combination of SEM and the Mahalanobis-Taguchi System to show that increases in household education level are correlated with reduced overall household health burden [40].

Considering these results, and the results of the seminal Whitehall studies [41], [42] which showed that socioeconomic status has a strong impact on health, we must conclude that we can achieve gains in household health not only by directly providing interventions like water filters, but also by 1) improving household education and 2) improving socioeconomic status.

### 2.5. CROWDSOURCING-BASED EMPLOYMENT AS AN ALTERNATIVE APPROACH

With the above lessons in mind, we pivoted to a drastically different approach to engineering development practice, focusing on employment to increase SES instead of directly providing technological solutions. In 2011, we began a pilot project to test the feasibility and investigate the impact of providing employment through a crowdsourcing work platform. Crowdsourcing is a novel labor organization paradigm which "represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call" [13]. Crowdsourcing is a "high growth, early stage industry" [32] that has the potential to radically transform the way work is done in the future (for a thorough review and a vision of the future of crowd work, see Kittur et al., 2013).

Seven villagers from Kamuga (site of the EWB project) were recruited to perform crowdsourcing tasks with a rate of pay of approximately six dollars per day. This compared favourably with the prevalent rate of pay of approximately one dollar and fifty cents over the same period. The workers were provided with four laptop computers, recharged using a portable solar array, and connected to the internet via 3G wireless modems. Each worker reviewed images collected from peer reviewed, archival biochemistry journals as part of a data mining project, paid for by the bioinformatics researchers who received the data [43]. Over a period of approximately two months, the workers earned a total of two thousand dollars with each worker receiving a fraction of the total in proportion to the number of images reviewed.

Approximately six months after payment, five of the seven workers were interviewed and asked to recall how the funds had been spent. Figure 2 provides the responses of each worker normalized to the total amount of income received. Each worker reported that more than 50% of his or her income was spent on education. These results support the hypothesis that access to income can result in improved education for villagers. The remainder of the income was spent on basic needs such as food and clothing, productive investments in small businesses or farming, and goods such as kitchen pots or radios. It is important to note that under this arrangement of employment for income, the liquidity of money allows individuals to meet their most pressing basic needs and make investments in their futures in the ways they find most compelling.

We currently believe that crowdsourcing employment is the most promising approach for reducing poverty quickly and at scale. This approach addresses the root cause of poverty (lack of money); it creates value for both workers and requesters of work (as opposed to aid, which relies on donations and merely transfers value); and it utilizes virtuous market incentives that align with social good. Access to the necessary computing infrastructure poses a challenge, but cybercafés, microfinance, and smartphones provide realistic options to lower the barrier to entry. The crowd work industry is large and growing; disbursements to workers totalled \$2 billion in the first decade of the industry's existence [31]. The biggest

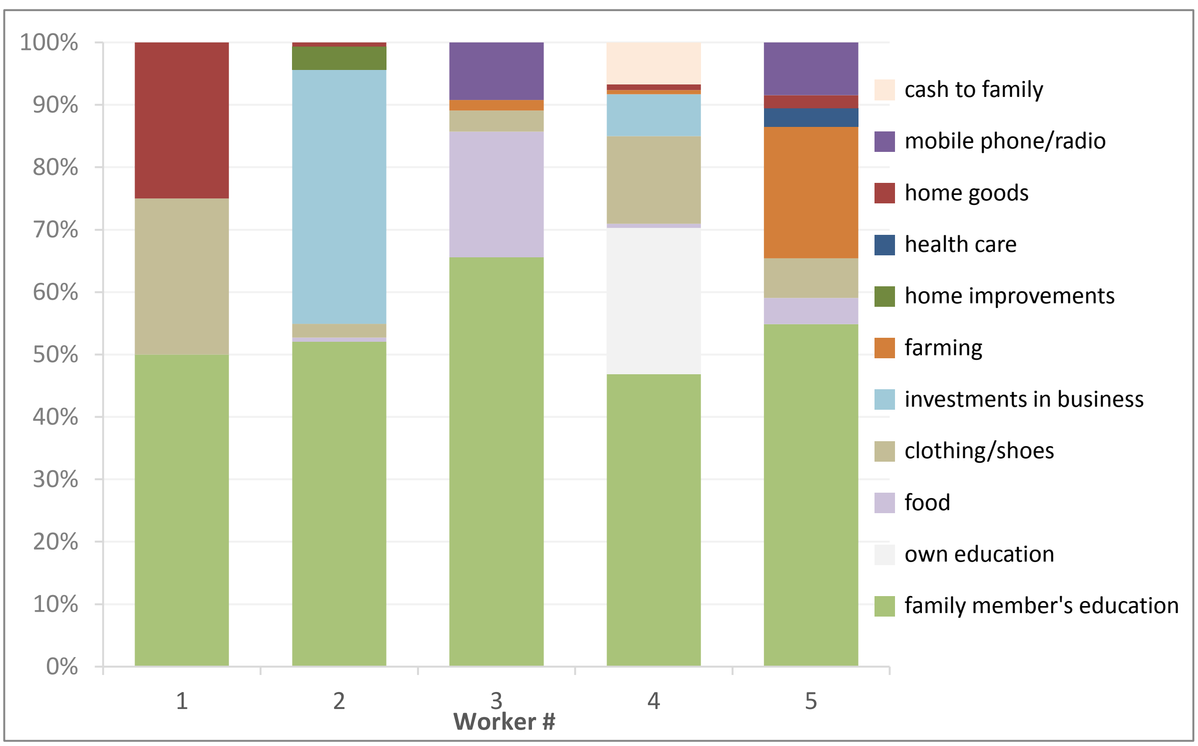



**Figure 2 Distribution of spending for 5 crowdsourcing workers in a Kenyan village.**

challenge is ensuring that crowdsourcing work is made widely available so that workers in developing countries have the opportunity to participate.

### 2.6. CONCLUSIONS

Development work is a significant challenge that has beguiled practitioners for decades, so we must think critically about our approach and learn from experience, lest we fall victim to the mistakes that have plagued the past. It is understandable that environmental engineers, with their water-related expertise, would gravitate toward water technology solutions to the poverty problem – "when holding a hammer, everything looks like a nail". However, this is not an acceptable justification to keep doing the same things without careful evaluation of the results. Over the course of several decades of evidence, it is not clear that the old approach is effective, and so new methods are required.

Through the experiences described above, we have come to believe that the most effective way for engineers to effect sustainable development is to provide employment opportunities (specifically via crowdsourcing). We would go so far as to say that, knowing what we know now, if we had the opportunity to go back 8 years, we would work on crowdsourcing employment, and not EWB projects. The incentives and the liquidity provided by income earned through employment represent a paradigm-shifting improvement over the old way of providing development aid.

## 3. Crowdsourced Human Computation Fights Poverty And Enables Novel Data Processing (Manuscript II, submitted to Science)

### 3.1. Abstract

*Unemployment in developing countries contributes to the persistence of poverty. However, crowdsourcing can potentially provide work to large numbers of people globally who lack specialized skills. Crowdsourcing also enables novel types of data processing, which we demonstrate with a bioinformatics data mining application. We investigate the feasibility of connecting people in rural areas to crowdsourcing employment with a pilot project and assess the way they use the resulting income. We find that workers in a rural Kenyan village are able to complete several example tasks, including approximately 100,000 bioinformatics image classification tasks. The income they received was spent primarily on basic needs, educational expenses, and productive investments. If such crowd work can be made available in sufficient quantity, this approach will dramatically reduce global poverty.*

### 3.2. Introduction

One of the great challenges of global economic development is the poverty trap [33], [44], created by low income and mutually reinforcing deficits in basic needs like access to clean water and health care. Globally 2.4 billion people still lived on less than $2 per day in 2010 [45]. In [44], Sachs argues for external aid to help the poor start up the ladder of growth. Easterly argues instead for staying out of the way, and letting market incentives drive economic growth [46]. As a middle path, we can both help people to start up the ladder, and avoid the backwards incentives created by aid handouts, by providing employment opportunities to the extremely poor. Recognizing that moving people out of the poverty trap ultimately requires that they find sustainable livelihoods, in 2008 the United Nations added Millennium Development Goal 1b: "Achieve full and productive employment and decent work for all, including women and young people".

Prior work [39] has also found that the effects of straightforward development interventions to meet basic needs like clean drinking water (the sorts of things Sachs promotes) are nonetheless modulated by factors such as socioeconomic status. This leads to the unavoidable conclusion: there are (at least) two paths to the goal of improving access to clean drinking water: provide water treatment technologies directly*, or improve socio-economic status*. Here we focus on the latter approach.

Meanwhile, recent work on crowdsourcing shows that, rather than being a novelty limited to simplistic tasks completed for pennies, crowd work, broadly envisioned, encompasses the future of distributed online work [30]. What the future of crowd work will look like will be determined by choices made in the present about access, ethics, and architecture. Of particular interest here is the extent to which crowd work is made globally available, especially in emerging economies. It is clear, however, that the concept of crowd work – distributed, asynchronous, scalable, on-demand – has significant practical and commercial appeal, and is here to stay. A 2009 market report estimated that in the previous 10 years, 1 million crowd workers had earned a total of up to $2 billion [31]; while a 2012 market report estimated the revenues of 15 crowdsourcing providers at approximately $400 million, with growth of 75% over the prior year [32].

Human computation is a type of crowdsourcing in which the tasks are repetitive and computation-like (people can be thought of as "processors" in a distributed computing system [16], and the tasks typically are hard for computers but easy for people). Human computation in various forms has been used to transcribe text [17], label images [18], and predict protein conformations [19]. Crowdsourced human computation allows simple process steps in a knowledge work assembly line to be distributed and performed by a global workforce. A key feature of this arrangement is that the individual tasks are simple enough that they can be completed by nearly anyone, with only minimal required training.

The intersection of price point and required skill level involved in repetitive human computation is well matched to employing workers in emerging economies. Consequently there is a significant potential for poverty alleviation *if the work can be accessed and completed by workers who are experiencing poverty*. Some previous work has investigated aspects of crowdsourcing employment opportunities in developing world contexts: Khanna et al. [47] studied the demographics of Indian workers on Amazon's Mechanical Turk platform and found a median annual income of $2700, and that 92% of survey respondents had a PC and internet connection in their homes, suggesting that most workers already have reasonably high socioeconomic status. They also studied the usability of Mechanical Turk and found significant barriers for first time users in India. Narula et al. [27] studied mobile phone-based word recognition tasks as an alternative to desktop PC-based interfaces, and found success with a small pilot group in an urban setting. Samasource [48] is a nonprofit organization that provides business process outsourcing services, partnering with in-country businesses to provide fully managed data processing centers staffed by disadvantaged workers. Recognizing that even within those countries classified as "developing countries," there exists a continuum of poverty, it remains an open question to what extent crowdsourcing can be leveraged to benefit the most severely impoverished areas. Additionally, to date no study has reported on the economic impact on human computation workers in developing regions.

The objective of the present study was to determine the feasibility of hiring people in rural Kenya to do human computation work for an example data analysis project (particularly, bioinformatics image processing) and to investigate how they spent their earnings. Three phases of work are reported here: feasibility testing, pilot project implementation, and follow-up assessment.

### 3.3. Feasibility testing

The initial feasibility test was undertaken to determine if 1) the price points in a rural setting were appropriate and 2) if the residents were able to work effectively with computers. The location of testing was the community of Kamuga, located within Nyakach constituency, in the Nyanza province of southwestern Kenya. The community of approximately 1000 people is located 60km by road from Kisumu (see map in Fig. 1). Most people living in the community participate in subsistence farming. A 2005 report estimated 64% of Nyakach residents lived below an approximate $1 per day poverty line [35]. Some community members live and work in another city, and send remittances home to their families. Employment opportunities within the community of Kamuga are limited: people may retail household necessities (foodstuffs, paraffin for lighting, soap, etc.) or provide manual labor to their neighbors, but the market for these activities, which is composed of other mostly subsistence farmers, is small.


**Figure 3 Map of Kenya showing the location of Kamuga, where the study activities were undertaken.**

A resident of Kamuga might hope to find work as a day laborer for 100-200 KSH per day ($1.18-$2.35, nominal exchange rate of 85 KSH per USD), but such work is scarce. Alternatively, for example, a person might find work as an untrained teacher (if he or she has not been to teacher's college) at the local primary school, earning 100 KSH per day. Attending teacher's college (at a cost of approximately 90,000KSH, or $1,058 per year for four years), might lead to steady employment at one of the local schools with earnings of 1000 KSH per day. Unfortunately, this investment is out of reach for most.

Seven residents of Kamuga were selected to comprise a variety of ages, occupations and prior computer experience; details are listed in Table 3. All participants had completed high school but no further schooling. The participants were screened for basic manual dexterity and visual capabilities before beginning any computer work.

| Sex | Age | Occupation | Prior Computer Use? |
|---|---|---|---|
| M | 21 | Untrained primary school teacher | Yes |
| M | 21 | Untrained primary school teacher | No |
| M | 46 | Unemployed | Yes |
| F | 19 | Unemployed (housewife) | Yes |
| F | 19 | Unemployed, occasional small business | Yes |
| M | 35 | Occasionally employed manual laborer | No |
| F | 25 | Owner of small fish-selling business | No |

**Table 3 Study participant demographics. The participants were chosen to represent diverse occupations, ages and computer experience.**

To facilitate the computer work, two Asus EEEPC netbooks were brought (purchased in the USA) along with three folding solar panels. Two additional netbooks were also purchased in Kenya. A car battery and USB 3G modems were purchased in Kenya. Kenya's extensive mobile phone network provides sufficient connectivity for these 3G modems. The total initial infrastructure cost was approximately $2000.

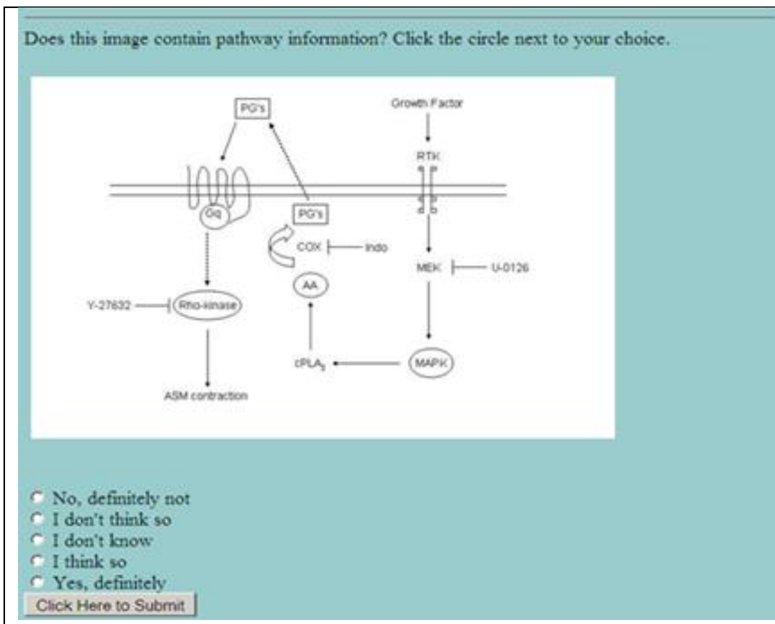
As an example of a visual/spatial data processing task, the participants were first tasked with playing through the initial training puzzles of the crowdsourced protein folding game Foldit [19]. Foldit players manipulate the shapes of proteins to generate improved predictions for the proteins' natural conformations. The current protein configuration is scored according to an energy function related to the protein's shape, and players manipulate the shape to improve the score. All participants were familiar with proteins from a high school biology course. After a brief introduction to Foldit, they were able to understand the task of changing the shape to improve the score, and then proceeded to the training puzzles. They were able to complete an average of 18 puzzles over a span of 8 hours.

It is encouraging that even people who had little or no experience using computers were able to successfully learn to complete visual-spatial tasks like Foldit using netbooks. Further, our interviews with residents demonstrate that what constitutes a "good wage" to people in Kamuga also constitutes a very "low cost" for purchasers of human computation services (several dollars per person-day of labor). Note however that this is not a "race to the bottom" of wages, or a "digital sweatshop", because of purchasing power parity. In summary, the initial feasibility testing shows that this is a promising approach.

### 3.4. Bioinformatics pilot project

An example of a paid human computation project was selected for work in Kamuga and a simplified web platform called PulaCloud[1] was developed. The example was chosen to be another representative human computation task: image classification. In this case the specific task was identifying representations of biochemical pathways in figures from medical journal articles; this processing step is the first step in a crowdsourcing workflow to annotate entities and relationships depicted in those pathways for computational biology data mining. This is an example of a class of visual human computation problems in which the task is to classify or categorize an object in an image, and where computer vision or machine learning would not be sufficiently accurate. With simple instructions, however, people can be trained to do the classification. Further details on this project are provided in Figure 4.

For their work on this image classification project, the workers in Kamuga were paid $2000, divided among the seven workers according to the number of responses submitted (approximately $0.02 per response, similar to a wage one might pay on Mechanical Turk). The workers made approximately $5.60




Workers used a simplified human computation platform (at left) to classify the images. A minimum of three answers were collected for each image and majority rule was used to determine the final answer. Workers were provided with training examples and then completed a qualification test before beginning work.

A total of 97,137 responses were collected for 28,481 images. Of these, 8,405 images were classified as pathways and will be further annotated for bioinformatics research.

Example image from [77].

**Figure 4 Example human computation project**

[1] The "pula" is both the currency of Botswana and the Tswana word for "rain".

(476 KSH) per day of work, significantly more than the other local employment options. The payment received by each worker ranged from $160 (13,600 KSH) to $514 (43,690 KSH) depending on the number of responses. At peak capacity the workers were completing over 9000 tasks per day.

A subset of 10% (2848) of the images were classified by an expert and used as a "gold standard" for measuring the accuracy of the workers in Kamuga. The overall accuracy was 92%; additional accuracy metrics are presented in Table 4 Accuracy of 3-response majority vote scheme on example image classification task.. The results on this task are encouraging; however additional improvements in accuracy may be made with improved training materials and more advanced quality assurance processes (see [49]–[53]) which are currently being implemented in the PulaCloud platform.

| | **Classification accuracy metric** | | |
|---|---|---|---|
| | ***Precision*** | ***Recall*** | ***Overall Accuracy*** |
| Definition | number of true positives (images correctly classified as containing pathways) divided by the total number of images classified as containing pathways | number of true positives divided by number of images actually containing pathways | fraction of correctly classified images |
| Result | 0.77 | 0.94 | 0.92 |

**Table 4 Accuracy of 3-response majority vote scheme on example image classification task.**

### 3.5. Impact Assessment

Approximately six months after payment, five of the seven workers were interviewed individually about how they spent their income from the project (the other two workers had found other employment in the meantime and moved elsewhere). During the interviews the workers provided amounts spent on each of 33 grouped categories. As a check on the accuracy of their recollections, we consider that 1) their answers for each of the categories were given in real-time in a conversational format, without pen and paper or calculator in hand to check their sums, and 2) for each of the five workers, when we later summed across categories, the sum across all categories was within 10% of the actual amount they received.

Each worker spent all of the income within two weeks after receiving it; this is not surprising in light of their ongoing unmet basic needs associated with living in extreme poverty. Figure 5 shows the distribution of the spending across four major categories of spending formed by groupings of the 33 subcategories for all workers, and Figure 6 shows a more detailed breakdown into eleven categories for each worker.

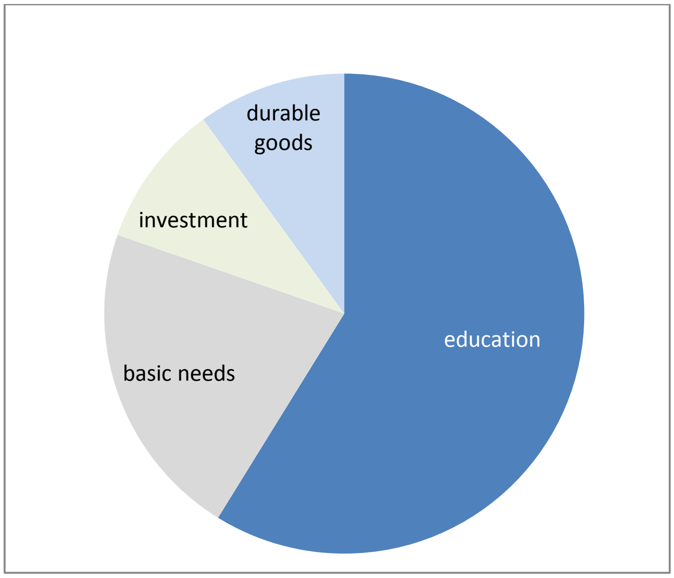



**Figure 5 Distribution of spending across broad categories. The largest share of earnings went to education. The remainder went to meeting immediate basic needs, like food and clothing; making useful investments for the future, like small business expansion and livestock; or goods like pots and radios.**

The workers spent the majority of their income on meeting basic needs or on productive investments. Most striking is that each worker spent half or more of the earnings on school fees for either him/herself or


**Figure 6 Detailed breakdown of spending across eleven categories for five workers.**

one or more family members (either a child or sibling), highlighting the value that the workers place on investments in education.

Several workers also indicated an interest in using future income to purchase additional computers and recruit more employees to expand their capacity to provide human computation services. Though the netbooks, solar panels and modems were loaned at no cost to the workers for this research, the costs of these items in Kenya can be brought within reach of workers with the help of microfinancing services, and their income from the work can be used to pay off the loans. In this case, the entire up-front investment in infrastructure was equal to the gross wages for one project, spanning approximately 3 months' time. Low-cost tablets, smartphones, and cybercafés provide additional options for computer access to the motivated individual. Internet access is another challenge in some areas, though it is rapidly expanding, and large firms like Google are making targeted investments in internet access for the developing world [24].

### 3.6. Discussion

The income from employment in this human computation project was put to use in a variety of productive ways to meet basic needs and make investments for the future. This suggests that such employment could be a very powerful tool for fighting poverty. The diversity of spending priorities beyond education also underscores the usefulness of an approach that transfers liquid assets in exchange for work;

it allows individuals to address the shortfalls in basic needs that they find most pressing, or make investments in their futures in the ways they find most promising. Contrast this with an economic development intervention in which a single need is addressed for a whole community (e.g. village-scale water access, sanitation, or transportation). With further scaling up, it is possible that our approach could enable a community to bootstrap its way out of extreme poverty. Moreover, this approach is mutually beneficial: while the workers get much-needed income, the requesters of the work get valuable data processing services. This is a notable improvement over the traditional aid model of a zero-sum transfer of value.

In order for this approach to impact a significant number of people in developing countries, the amount of human computation work available to them must be increased. There are two components of availability: workers' ability to access platforms, and the amount of work on those platforms. Ability to access platforms can be addressed in a number of ways as discussed above, but *access to platforms is only valuable if there is work to do on the platforms*, i.e. the expected return on investment for workers must be high enough for them to make the necessary investments. Thus the single most important factor determining whether this approach will be successful is how much work is available to disadvantaged workers.

Increasing the overall demand for human computation services (by awareness and education of potential users of human computation) is one option. The utility of human computation for data analysis and problem solving has been thoroughly demonstrated [30], though it is still not widely used either in scientific research or commercial settings [31][25]. We and others [26] believe this is due to an awareness gap and to obstacles in specification and setup of a human computation project. Additionally many current data analysts (we use this very broad term intentionally) are not accustomed to seeking out solutions that involve large-scale, brute force human effort. Hence the crowdsourcing industry is still a "*high growth, early stage industry*" [32]. The other option for increasing the amount of work available to disadvantaged workers is deliberately increasing the fraction of human computation work that is sourced in emerging economies. This would require purchasers of crowdsourcing labor to select a platform that makes work available in emerging economies in preference to one that does not. Increasing the amount of available work is the biggest obstacle to further adoption and scale-up of this approach and the biggest opportunity to help it succeed.

### 3.7. Conclusions

This study has demonstrated that it is feasible to employ workers in rural areas in developing countries with human computation tasks. One batch of nearly 100,000 tasks was completed, resulting in the transfer of $2000 to seven people in a rural community experiencing extreme poverty. The income was used in a variety of productive ways, indicating that this is an effective poverty alleviation strategy. The necessary infrastructure was provided for this pilot project; future work is needed to determine how to provide access to this infrastructure in a sustainable and scalable way. Most importantly, however: we urge the rapidly expanding crowdsourcing industry to take care to make work as widely available as possible so it can be completed by workers in developing countries. If these issues can be addressed, this new approach has the potential to lift many people throughout the world out of poverty, while simultaneously enabling new methods of data processing.

## 4. No really, (crowd) work is the silver bullet (Manuscript III, presented at Humanitarian Technology: Science, Systems and Global Impact 2014, HumTech2014)

### 4.1. Abstract

Humanitarian assistance has been on the global conscience for approximately 70 years (since WWII), and yet in 2010 2.4 billion people still lived on less than $2 per day. As Easterly has pointed out: to see where we went wrong, just look at the incentives. To create true sustainable economic change requires realignment of incentives, particularly the incentive to work and invest. Employment is fundamentally required, and crowd work is the current best hope for providing that employment quickly, with global reach, and at scale. This approach is grassroots, bottom-up, and puts the income directly in the hands of people who need it. Further, it leverages the natural, inherent incentives embodied in capitalism (workers work to create value and get paid, employers want to minimize costs of labor) to shift as much work as possible to the places where it will have the most beneficial impact. We present an analysis of global trends supporting crowdsourcing as a solution, and the results of a pilot project in a rural Kenyan village which demonstrates that this approach is an extremely promising way to meet basic needs to promote economic growth.

### 4.2. Introduction

In 2010, 2.4 billion people worldwide still lived on less than $2 per day [45]. Addressing this issue has been, if not a global priority, at least on the global conscience for approximately 70 years (since the end of WWII). Throughout those 70 years many different approaches have been tried. There continues to be vigorous debate about the merits of different approaches: e.g. Sachs argues for the effectiveness of foreign aid and thus for increased aid expenditure to start people up the ladder of growth [44], Easterly argues for reducing aid in favor of approaches that align individual incentives with large scale goals [46], and Banerjee and Duflo argue that individual interventions should be evaluated on a case-by-case basis [33]. Each point of view has its merits; instead of weighing in on which is best, here we present an approach that is in accord with the major ideas of 1) a jump start up the ladder of growth, 2) properly aligned incentives, and 3) measurement of results.

Any reasonable view of sustainable development includes a picture of economic self-sufficiency; for the poor to become (and stay) not-poor, most would agree that they need opportunities for productive, value-creating employment. They may or may not need bed nets, toilets, or cell phones, but it is certain that they need jobs. Many popular development interventions, including water treatment and distribution, improved healthcare, and education, are often justified on the grounds that they allow the poor to be more productive, via more time, energy, or skills. It is in this sense that we assert that "work is the silver bullet". It is almost tautologically true that doing valuable work and getting paid for it results in economic growth.

For clarity we should make several disclaimers. Here we are targeting systemic or structural poverty, caused by factors external to an individual and excluding poverty caused by, e.g. mental illness (this is not to say that this sort of poverty is "less bad" – only that it is not the focus of this paper). We also recognize that economic growth in the sense of increased income is not the ultimate goal, but rather a contributor to a broader type of human development. We would argue, for example, that improved health is a terminal goal, and that instead of improving health to improve productivity, we aim to improve socio-economic status in order to improve health (as demonstrated by the seminal Whitehall studies [41], [42]). Finally, while we will argue that "crowd work" (that is, broadly envisioned crowdsourcing-based work) is a nearly universally applicable tool, we expect it to lead eventually to diversification of employment.

### 4.3. Why "crowd work" works

#### 4.3.1. First – why work?

Beyond being closely related to economic growth, providing opportunities to work for income eliminates the damaging incentives created by aid handouts. When the path to development goes through employment, there is an incentive to use resources to invest in the future: e.g., education, physical capital, infrastructure. Further, when an employer pays an employee for labor, both sides receive something of value, whereas aid relies on the generosity of donors (or perhaps their extremely low discount rate for the time value of money). To give an indication of the scale of resources at play, consider Figure 7, which compares Foreign Direct Investment (that is, private investment by businesses) and Official Development

Aid from all sources into developing countries over time [7][8]. In 2010 FDI was approximately seven times as large as ODA, and growing much more quickly.

Finally, any employment scheme that puts money in the hands of people who are experiencing extreme poverty allows those people to invest it in the ways that they find most beneficial, using their local knowledge and based on their own priorities. Earned income can be used to pay for water treatment, food, latrines, bed nets, health care, etc. The liquidity of income is a distinct advantage over in-kind transfers. The question of how the income is actually used, for a pilot study of the crowd work approach, is addressed below.

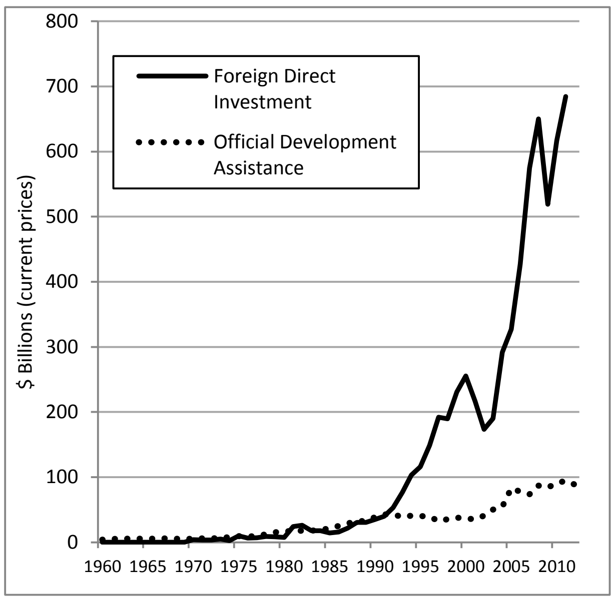



**Figure 7 Comparison of the size of foreign direct investment (FDI) and official development aid (ODA) over time. Beginning in the early 1990's FDI began to rapidly outpace ODA.**

### 4.3.2. What is "crowd work"?

Crowd work is a broad term that describes a type organization of labor in which laborers are loosely affiliated with firms, and work of many types is widely distributed. It is an extension of "crowdsourcing," which, as defined by Jeff Howe, "represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call" [13]. Recent work by [30] has highlighted the benefits of this system of labor organization for both firms and laborers and laid out a future vision and research agenda for a system that is advantageous to all. Crowd work encompasses both "human computation" or "microtasking" – work in which the tasks are very small, short, and simple – and more complex and highly skilled work such as graphic design, computer programming, and technical research and development [14][15].

#### 4.3.3. Unique features of crowd work make it especially appropriate as a poverty-fighting tool

Crowd work is especially well-suited to fighting poverty for a number of reasons. It is inherently distributable to any place with internet access (and while many areas of the world do not yet have access, large firms such as Google are making targeted investments to increase connectivity in developing countries [24]). At the human computation or microtasking end of the spectrum of task complexity, the work requires no specialized skills and minimal training, making it broadly accessible to almost anyone. Further, as skills are developed, crowd work provides opportunities for more engaging and demanding work (similar to the development of the information technologies services industry in India). On existing crowd work platforms such as Amazon's Mechanical Turk, microtasks are often priced at a few pennies each, allowing workers to earn on the order of several dollars per hour. While many US-based "Turkers" complain about the low rates, for someone who otherwise would make $1-$2 per day for difficult manual labor, these wages are attractive.

The crowdsourcing industry is a "high growth, early stage industry", with revenue growth of 75% 2010-2011, and total 2011 revenue from a sample of 15 crowdsourcing companies of $375 million (this is revenue to the companies, and is only an indirect indication of disbursements to workers; many such companies charge an overhead fee on top of each transaction) [32]. A related industry, "impact sourcing", in which traditional business process outsourcing work is sourced in developing countries for social benefit, was recently estimated at $4.5 billion in size, with projected growth to $20 billion by 2015 [56].

Finally while access to the necessary electricity, internet access, and computing resources presents a challenge in many (especially rural) areas, it is not insurmountable. MobileWorks [27], mClerk [28], and TxtEagle [29] are all examples of crowdsourcing systems built to leverage mobile phones (basic phones or smartphones). We believe that if there is enough available crowd work, potential workers in developing countries will be able to make the necessary investments to participate (possibly via cybercafés, microfinancing, cooperatives, or other arrangements).

### 4.4. Results from our pilot project and income survey

To investigate the feasibility of providing crowd work in rural areas of developing countries and assess the impact it would have, we did a pilot project in the community of Kamuga, Kenya. Most of the 500 residents of Kamuga live below the $2 per day poverty line, and participate in subsistence farming. We recruited 7 people (4 males, 3 females) ranging in age from 19 to 46, 4 of whom had previous computer experience and 3 of whom had none. We developed a simplified crowdsourcing platform called PulaCloud, and tasked the workers with classifying approximately 28,000 images from biomedical research articles as to whether or not they depicted a biochemical pathway. The image classification task is for a bioinformatics research project to make pathway figures searchable and data mine-able. The workers completed approximately 100,000 tasks and were paid $2000 for the project, divided among the seven according to how many tasks they completed. Six months later we interviewed 5 of the 7 workers (the others had moved away from the community) about how they spent their income. Figure 8 shows the distribution over 4 major categories formed by grouping the 33 subcategories from the survey. The workers spent over half of the money on educational expenses for themselves or a family member; the rest was spent on basic needs (food, clothing, health care), investments in small businesses (e.g. selling fish in a kiosk or paraffin for lighting to their neighbors), and goods like pots and radios. We find the choices that workers made about how to spend their income extremely encouraging.

Finally, we would like to note that PulaCloud is organized as a for-profit entity. The profit incentive for PulaCloud means increasing the throughput of crowd work on the platform (which means increasing the payments to workers) as well as minimizing costs (which means seeking out those workers experiencing

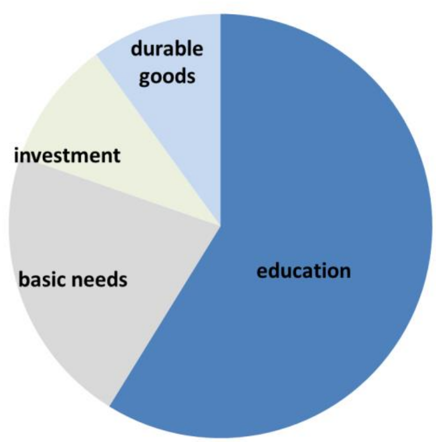



**Figure 8 Distribution of spending across 4 major spending categories formed by the grouping of 33 subcategories. The workers spent the majority on education, and spent nearly all of it on productive investments for the future.**

the most severe poverty). This is not a "digital sweatshop," however; it is simply a case where the individual incentive is closely aligned with the collective good.

### 4.5. Conclusion

We believe that crowd work can and should be a major contributor to the elimination of global extreme poverty. It has a number of significant advantages as an approach to economic growth, and has the potential to scale up to considerable size. Not only this, but there is no need to wait - it can be applied starting right now. The main impediment is the fact that much of the currently existing crowd work is not available to workers in developing countries because the platforms either actively block international workers from registering (Mechanical Turk) or do not have payment options for many countries. This despite the fact that Kenya, for example, has a very advanced mobile payment system in MPESA.

We have shown with our pilot study that it is both feasible and advantageous to employ workers in developing countries in crowd work. We recommend 1) further scaling up of the crowd work approach and 2) concomitant additional study of the effects of crowd work employment on poverty.

## 5. What Difference Does the Device Make? Crowd Work on Computers and Phones (Manuscript IV, submitted to HCOMP 2014)

**Andrew Schriner[1], Raja Bolla[2], Christopher Brown[2], Sriram Chellappan[2], Daniel Oerther[2]**

### 5.1. Abstract

How effective are smartphones for completing crowdsourcing and human computation tasks (i.e. crowd work)? With global smartphone adoption rising (especially in emerging markets), distributing tasks on phones becomes more appealing both for workers and requesters. Can tasks be transplanted as-is for completion on phones, or must special considerations be made for the phone interface? In this study we conduct an experiment comparing computer and smartphone user responses for three types of crowd tasks. We recruited 40 crowd workers and randomly assigned them to complete 852 total tasks on a computer or phone. We analyze differences in timing, duration, accuracy, and satisfaction between the two groups. Results show that workers on phones are generally slower and less accurate. Worker satisfaction, however, varies according to the type of task. Finally, we discuss implications for the design of crowd work platforms.

### 5.2. Introduction

Crowd work presents exciting opportunities for the global distribution of income-earning opportunities and the recruitment of appropriate labor for specific tasks. To achieve its potential, however, crowd work must be widely accessible, and designed appropriately for the medium of distribution. To date most crowd work has been distributed on platforms with a focus on full size desktop- or laptop-based interfaces. In the meantime, “responsive design” – web design that adjusts to accommodate users on different size devices – has been gaining traction. Further, the cost barrier for many potential crowd workers throughout the world makes full size desktops or laptops an unrealistic option for accessing crowd platforms. For example, in Kenya, the cost of a low end laptop is approximately $250[2], while an Android smartphone costs approximately $50 (both figures purchasing power parity)[57].

Previous work by [28] and [29] has shown the effectiveness of phone based crowdsourcing for simple text tasks over SMS, and even with small images, in India and Kenya. Others have tested the usability of feature phones and a basic web browser for completing OCR tasks in India [27] and found that the simple interface was effective for this type of task. On the other hand, [47] found that the Mechanical Turk interface (on a desktop) was too complicated to the point of being entirely unusable for new users without significant computer experience. In [58] the same authors lay out an agenda for maximizing the benefit of crowdsourcing for potential workers in developing countries, including further expansion of platforms into mobile phone interfaces.

With these opportunities in mind, we ask: Are there issues or opportunities that designers of crowd work platforms need to consider with regard to workers on mobile phones? How does the technology available to a worker affect how they interact with an online human computation marketplace? Are there differences in behavior between desktop/laptop users and smartphone users that should be taken into account in the design of a human computation marketplace?

In order to push the boundaries of phone-based crowdsourcing, we compare desktop/laptop users (for brevity, “computer” users) and phone users across several different task types representing a diversity of interface interactions. We investigated several possible differences in behavior among the different device user groups. In this paper we address the following questions:

[2] From http://www.jumia.co.ke/notebooks/, and corroborated by authors’ experience in stores in Kenya.

1. Do phone users tend to work in shorter bursts than computer users?
2. Do computer users tend to work around certain times of the day, while phone users work any time? Perhaps we can get lower latency with more phone users in the crowd.
3. Are there differences in accuracy across task types between user groups?
4. Is there a difference in how long a user takes to complete a task between phone and computer users?
5. Are there differences in worker satisfaction between the phone and computer groups?

### 5.3. Methods

We recruited 40 participants from Rolla, MO, affiliated with the Missouri University of Science and Technology (37 students and 3 staff members). Participants were randomly assigned to either the phone or computer group (20 per group), and any participant in the phone group who did not already own a smartphone was provided one. The participants were instructed to use only the device type to which they were assigned to complete the tasks (and this was verified against the user agent reported by their browser when submitting results). To address differences between computer and phone users, we prefer randomization rather than a natural experiment in which we only record user agents, because it helps to eliminate group self-selection effects. Participants were paid $75 if they completed all of the tasks.

The tasks were completed using a crowd platform called PulaCloud, which has been used in previous studies of crowd work in developing countries [43]. PulaCloud uses a simple responsive grid interface which presents tasks in a horizontal format for wide screens and a stacked format on narrower screens. See Figures 9 and 10 for examples of one task in the narrow and wide formats, respectively. When beginning a new task type, workers are first shown a set of instructions, and then presented with interactive training tasks. When a training task is answered incorrectly, the user is shown the correct answer and an explanation. Workers were able to redo the training tasks as many times as needed until they got 2/3 of them correct. At this point, they moved on to complete the rest of the tasks of that type.


**Figure 9 Streetview task on narrow smartphone screen.**

Workers completed three types of tasks chosen to represent a diverse set of interfaces and user interactions. All tasks are borrowed from the authors' other crowdsourcing projects. The Streetview and house data tasks come from air quality research on exposure to indoor and outdoor air pollution as a function of window opening, and the facial expression task comes from research on cultural differences in perception of faces.

- *Streetview task*: the user was shown an interactive Google Streetview window in an iframe. The view was directed at a particular address (there were 282 addresses for 282 tasks). The user was asked 1) if the view provided a clear view of the home; 2) if yes, they were asked if any windows were open; 3) if yes, how many. This task required making a judgment as to 1) whether or not all windows were visible enough to decide if any were open, and then 2) making a judgment as to whether or not the windows were open, and finally 3) how many windows were open. In the instructions and training tasks,

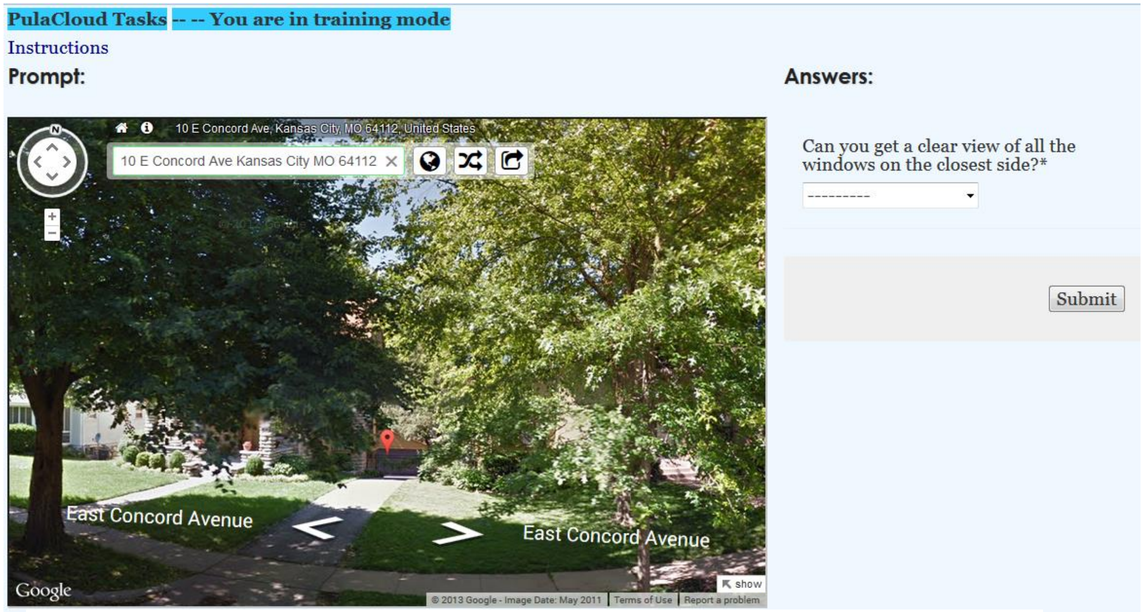



**Figure 10 Streetview task on wide screen; no scrolling necessary. Image from "http://instantstreetview.com/s/10 E CONCORD AVE KANSAS CITY MO 64112"**

the users were directed to use the Streetview controls (move location, change direction, and change zoom level) to get the best possible view of the windows on the home.

- *House data task*: the user was provided a link to the Zillow.com search results for an address and instructed to visit the page. The user was then asked to report the 1) "Zestimate," Zillow's estimate of the home value (NB: we did not include this data in the accuracy analysis because of ongoing updates by Zillow), 2) the square footage of the home, and 3) the number of bedrooms. This task required the user to locate and either type or copy/paste pieces of information from one site into another.
- *Facial expression task*: the user was shown an image of a person, and asked to report 1) if the person was displaying positive, neutral, or negative emotion (on a five-point scale) and 2) if the picture showed the person in a positive, neutral, or negative light (also on a five-point scale). For this task the only required user interaction with the interface was selecting the answer. The task required the user to make a subjective judgment about the emotion and the overall impression of the person in the image.

These particular tasks were selected to comprise a range of interface interactions from simple to complex, and answer types from binary to multiple choice to free input. Table 5 summarizes the salient characteristics of each task.

| Task | Input (prompt) | Question type | Number of tasks |
|---|---|---|---|
| Streetview | • Complicated and interactive interface<br>• Image input | • Binary multiple choice, numerical free-input<br>• Objective, but with uncertainty because of unobservable ground truth | 282 |
| House data | • Interface required navigating to external link and back<br>• Numerical input | • Numerical free-input<br>• Objective | 287 |
| Facial expressions | • Simple interface<br>• Image input | • Multiple choice, ordinal<br>• Subjective | 283 |

**Table 5 Characteristics of inputs and outputs for tasks chosen for this experiment**

## 5.4. Results and Discussion

Of the 40 participants, 18 completed all of the tasks (9 each from the phone and computer groups). Unless otherwise noted, we limit our analysis to the users who completed all of the tasks. The tasks were completed from November 13 – December 6, 2013. The remainder of this section is structured to answer each of the questions posed in the introduction.

### 5.4.1. Do phone users tend to work in shorter bursts than computer users?

To address this question we first must choose a definition for sessions. In the case of piecemeal crowd work, choosing a definition is challenging because workers may not give the tasks their full attention, perhaps watching TV or holding a conversation while working, and short lulls of inactivity are common. To address this, we plotted the pooled distribution of intervals between task submissions for all workers and all tasks, and found that a 300 second (5 minute) interval made a reasonable cutoff, capturing 97% of all between-task intervals. The following analysis was conducted using three definitions for a session: 1) any break longer than 5 minutes is the end of a session 2) any break longer than 10 minutes is the end of a session, and 3) any break longer than 5 minutes is the end of a session, unless the break is less than 10 minutes and the next interval is less than 5 minutes. The results for all three definitions were similar; we present the results for definition 3.

We investigate length of session both in terms of number of tasks per session and time duration of session. Both cases were modeled using a generalized linear mixed model with the individual participant as a random effect and group membership as a fixed effect. For tasks per session, we use a Poisson distribution for residuals and identity link function, and for time duration we use a Gamma distribution for residuals and identity link function. Our data contains 336 sessions (dropping the last session for each user because its length was truncated by running out of tasks); 177 for the computer group and 159 for the phone group. For the number of tasks per session, we find weak evidence that phone users work in shorter sessions (mean difference 28.7 tasks, $p=0.187$), while for time duration we find no difference (consistent with our finding, discussed later, that phone users on average take longer to complete 2 out of the 3 task types). See Table 6 for a summary of findings. A power analysis suggests that a sample size of 3x larger than ours would be required to detect a difference in number of tasks per session at alpha=0.05, holding variance and effect size constant. We plan to recruit more participants in the future to address this.

#### 5.4.2. Do computer users tend to work around certain times of the day, while phone users work any time?

Combining the sessions from all users in each group, we compare the distributions of session start times. Figure 11 shows the counts of sessions started each hour of the day. Kuiper's test (a variant of the Kolmogorov-Smirnov test which can be used on cyclical data) was used to determine if the counts come from two different distributions. We do not find a difference between the distributions ($V = 0.152$, $p=0.321$). Note however, that in the hours from 5-8am, 4 different workers in the phone group began sessions, while only two workers in the computer group began sessions, both during the 8am hour; and phone group workers began 22 sessions during the hours from 2am-8am, while computer group workers began 11 sessions during that time. It is possible that phone users are more likely to work overnight because a phone is more accessible than a laptop or desktop, but additional data is needed to make a clear distinction.

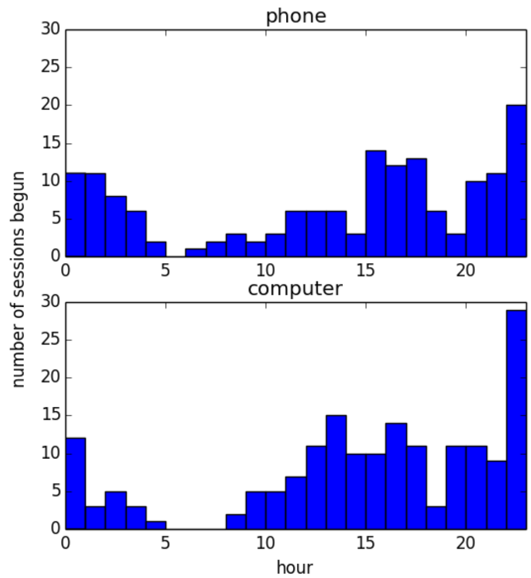


**Figure 11 Session start times for phone and computer groups. All times are Central Time.**

| | Tasks per session | Time duration of session |
|---|---|---|
| Mean difference | -28.7 tasks | 79.8 seconds |
| p-value | .187 | .827 |

**Table 6 Model coefficients and p-values for session length. Mean difference is computer group mean minus phone group mean.**

### 5.4.3. Are there differences in accuracy across the task types between user groups?

For each task, all user responses were binary classified, correct or incorrect, according to a gold standard. For the Streetview task, the answer to the “any windows open?” question was only used if the answer to the “is there a clear view?” question was correct, and likewise for the “how many windows?” question. For the house data task, the “Zestimate” was left out of the analysis because Zillow regularly updated that data and consequently there was no gold standard. For the facial expressions task, user responses were classified as correct if they were within one step in the 5-point ordinal scale (e.g. if the gold standard was “slightly negative”, then both “very negative” and “neutral” were accepted as correct answers).

To determine the effect of phone or computer group membership on accuracy, we use logistic regression on the binary correct/incorrect outcome. As before, group membership is modeled as a fixed effect and individual worker as a random effect. For each question and each task, with the exception of the "Can you get a clear view?" question for the Streetview task, phone users had significantly worse accuracy, at levels from $p<<0.001$ to $p<0.1$. The magnitude of the accuracy differences is enough to be of practical importance for crowdsourcing results. See Table 7 for details; the log odds are the coefficients estimated in the regression model, and mean accuracy is log odds converted to probability of a correct answer.

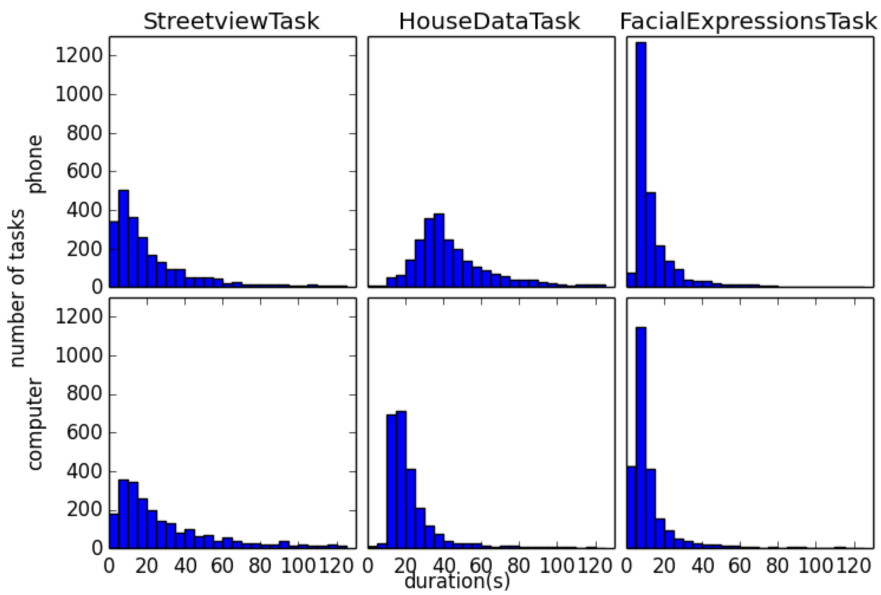
| Task | Question | n | Log odds | | Mean accuracy | | p- value |
|---|---|---|---|---|---|---|---|
| | | | Phone | Computer | Phone | Computer | |
| Streetview | Clear view? | 4961 | 0.65 | 0.76 | 0.66 | 0.68 | 0.631 |
| | Windows open? | 938 | 1.25 | 2.39 | 0.78 | 0.92 | 0.0088 |
| | How many? | 46 | -0.86 | -0.81 | 0.30 | 0.69 | 0.0844 |
| House data | Square feet? | 4982 | 2.57 | 4.70 | 0.93 | 0.99 | 2e-5 |
| | Bedrooms? | 4982 | 3.50 | 5.52 | 0.97 | 0.996 | 0.0018 |
| Facial expressions | Emotion? | 4958 | 2.47 | 4.15 | 0.92 | 0.98 | 0.0117 |
| | Overall light? | 5103 | 1.50 | 2.41 | 0.82 | 0.92 | 0.0157 |

**Table 7 Logistic regression results for accuracy as a function of phone or computer group and a random worker effect.**

#### 5.4.4. Is there a difference in how long a user takes to complete a task between phone and computer users?

We collected two kinds of data on task duration: 1) time between GET and POST requests to the server (i.e. the time from when the task was requested and when the result was received by the server) and 2) time recorded by a client-side Javascript timer that started when the page was fully loaded and stopped as soon as the "Submit" button was clicked. The Javascript timer was implemented to isolate the effect of page loading times from task completion times; however, the Google Streetview window used AJAX to load Streetview imagery and continued loading after the timer started. At the same time, it is possible for a user to begin mentally working on a task when the task is only partially loaded (before the Javascript timer starts); these issues make it difficult to separate page loading entirely from task duration. Finally, as with session duration data, task durations will be increased when a user is giving the tasks only partial attention, and it is not possible to identify from the duration data when this is occurring.

The results of the analysis are similar for both types of duration; Javascript timer data will be presented here. For the house data task and facial expression task, phone users took longer than computer users; for the Streetview task, computer users took longer. All differences are significant using a Mann-Whitney U-Test with p-values << 0.001, though the difference in the facial expressions task is small enough to be practically irrelevant for most applications. **Error! Reference source not found.** shows the median task durations and Figure 12 shows the histograms for each task and group.




| | Streetview | House data | Facial expressions |
|---|---|---|---|
| phone | 13.5 | 39.5 | 9.3 |
| computer | 22.7 | 18.7 | 8.1 |

**Table 8 Median task durations (in seconds) for each task type and group.**

### 5.4.5. Are there differences in worker satisfaction between the phone and computer groups?

We asked all workers to complete a survey about their experiences using the system, whether they completed all of the tasks or not. Of the 18 users who completed all the tasks, 17 completed the survey (8 from the phone group and all 9 from the computer group), and 5 workers who did not complete all the tasks also completed the survey. Only the users who completed all the tasks were asked about satisfaction on each task; users who did not complete all the tasks were asked why they chose not to complete them.

Computer group workers reported higher satisfaction on the Streetview and house data tasks; while phone group users reported slightly higher satisfaction on the facial expression task. When asked if they would be interested in completing work like this in the future (i.e. to make money, outside the context of a research study) 21 out of 22 respondents chose "Very interested" or "Somewhat interested" (18 chose "Very interested"). Respondents indicated 1) they were interested in earning income and 2) they found the tasks interesting, echoing previous research on Mechanical Turk [59]. In response to a question about desired pay rates, 17 of 22 indicated they would work on such a system if they could make $8 per hour (10 of 12 computer users, and 7 of 10 phone users).

### 5.5. Conclusions

We find some differences in behavior between computer and phone users in a crowdsourcing task system. Differences in duration are most likely due to aspects of the device interface; smaller screens make some interactions harder. Complicated interface interactions like those on the Streetview task, and navigating across multiple pages to complete a task, are made more challenging by the small touchscreen. When payment is made per time (rather than per task), requesters may want to consider the differences in speed of work introduced by the device type.

Similarly, differences in accuracy can be attributed to device and interface characteristics. The larger screen of a desktop or laptop reveals more detail in images than the smaller screen of a smartphone. Likewise the small keyboard invites typing errors in free-entry tasks. These differences demand additional attention from task interface designers if improvements in phone-based crowd work are to be made. Specifically 1) screen layout should maximize input image display size and 2) for numeric input fields, task designers should specify an input type that triggers the numeric keyboard on mobile devices.

It is possible that higher worker satisfaction on some tasks on the smartphone may lower workers' reservation wage for those tasks, allowing requesters to purchase crowd labor at lower cost. If the task interface reduces accuracy for phone workers, there may be a tradeoff between accuracy and cost; for other tasks there may be no tradeoff.

Ultimately, we can say that some tasks are notably better suited to the computer format, while others may be equally well suited to either, with appropriate design. As a corollary we can suggest that for a crowdsourcing platform that distributes tasks to phone users, minimizing complicated interface interactions will improve speed, accuracy and worker engagement.

At the same time, we find that workers are nonetheless able to complete all three task types on both devices, with mostly reasonable accuracy, and that they are satisfied with the experience.

Therefore we recommend further expansion of crowd work marketplaces into the smartphone format, with the caveats that only some tasks will be appropriate for completion on phones, and task designers may want to include some phone-specific considerations in their interface design. Tablets are another possible interface, and future work is planned to situate tablets on the continuum of devices. Improving the experience and accessibility of crowd work on these devices will be especially beneficial for increasing the global impact of crowd work opportunities.

# 6. Flowbuilder, the crowdsourcing stack developer's toolkit (Manuscript V, submitted to Computer Supported Cooperative Work, CSCW 2014)

## 6.1. ABSTRACT

Many crowdsourcing projects share a subset of architectural and functional design requirements, such as the need to create gold standard answers to a subset of tasks and segment the crowd by skill level. Multi-task workflows create additional demands, such as the ability to pipe results from one task to the inputs of another task. By identifying the common elements of crowdsourcing projects, we have factored these components out into a software toolkit for crowdsourcing application developers. By analogy to the web stack, we call this set of common components the “crowdsourcing stack”, and we present an implementation of this stack called Flowbuilder. We present three use cases of Flowbuilder for two purposes: two cases primarily focused on data processing, and one case for conducting experiments to study crowdsourcing itself.

## 6.2. INTRODUCTION

To go from initial ideation to successful implementation of a crowdsourcing project requires a significant amount of software development work. This often also requires the carefully considered implementation of crowdsourcing-specific paradigms such as gold standard answers. While there is tremendous diversity in the uses of crowdsourcing (and here we include both volunteer and work-for-pay arrangements), many such projects share a similar set of core crowdsourcing-related functionality. To date no software packages are available which both encapsulate this shared functionality and focus on the practical issues that confront crowd-backed application developers in their efforts to extract value from crowdsourcing. The creation of a crowdsourcing software developers’ toolkit would thus reduce the time and effort required to get a new crowd project off the ground while reducing the need for individual developers to learn all of the best practices that have been identified by crowd work researchers and practitioners.

By way of example, consider the common crowdsourcing features in these crowd projects. GalaxyZoo [60] tasks users with viewing images of galaxies and classifying them. Users 1) register and then begin with 2) instructions and 3) training examples, and then proceed to provide new classifications, which are 4) aggregated with responses from other users. TomNod [61] recruits volunteers to view satellite imagery and search for features of interest, such as flood damage or the wreckage of Malaysian Airlines Flight MH370. TomNod users 1) register and 2) view instructions before going on to tag features that are then 5) verified by other users. A tweet sentiment analysis task running on MTurk or CrowdFlower will typically include 2) instructions, 3) training examples, 4) redundancy for quality assurance and possibly hidden 6) gold standard tests.

It is an inefficient use of developer time to reimplement these features for every new project. Developers of web stack frameworks such as Ruby on Rails and Django recognized this general problem, and now these frameworks allow web app developers to focus on coding what is unique about their website, rather than the common, "boring" components. Further, the widespread adoption of these frameworks supports conceptual standardization of web apps along the Model-View-Controller paradigm [62], enforcing to some extent the adoption of best practices. With the development of a crowdsourcing app toolkit, we likewise hope to ease the burden on developers and encourage the adoption of best practices.

Crowd work represents a continuum of required skillsets from very simple microtasks to complex, expert work [14], [30], and a developer's toolkit should reflect this reality. Even within the realm of microtasking, crowd workers are heterogeneous in their skills and effective workflow design must take this into account. Further, for many tasks, workers' task-related skills are not an intrinsic, time-invariant property of the individual worker; rather, the human ability to learn and adapt presents an opportunity for crowd work requesters to train up a crowd of targeted, skilled workers.

A number of frameworks for programming with crowds have been presented in the literature, including CrowdForge [63], Turkit [64], and Jabberwocky [65]. All three rely on the conceptual model of "human computation" (i.e. humans as computers) which has significant drawbacks when applied to a broader definition of crowd work [30], [66], [67]. Additionally, the CrowdForge source code is licensed so as to prohibit commercial use, and Jabberwocky is not available for public download, preventing adoption of these systems for solving real business and research problems.

Importantly, Flowbuilder is designed primarily to be actually used in real-world applications by developers, not as a way to demonstrate concepts to the crowdsourcing or human computation research communities (NB - Flowbuilder is available for download at github.com/aschriner/flowbuilder and licensed under the GPLv3 open source license). Flowbuilder includes functionality to address a number of practical concerns related to getting the most value out of crowd work, including logic for training workers, the ability to patch crowd functionality onto existing legacy systems or databases, and native integration of a validated quality assurance system. Flowbuilder was designed with special consideration for the developer's mental models, in order to ease the cognitive burden of developing crowd-backed applications. The concepts and terminology align more closely with the paradigm of "humans using a web application" (accepting all of the usability challenges, nuance and frailty of that arrangement) than a machine assigning tasks to "processors in a distributed computing system" [16].

In this paper we present an analysis of the components of an archetypal crowdsourcing project, which we synthesize into a conceptual model of crowdsourcing from the developer's perspective. We call this model the "crowdsourcing stack". We then describe an implementation of a toolkit for building crowdsourcing projects which implements the components of the crowdsourcing stack. Finally, we describe several use cases where this toolkit has been used to implement various crowdsourcing projects.

### 6.3. The crowdsourcing stack

What are the individual components that make up an archetypal "crowdsourcing project"? In a broad sense, because crowdsourcing is inherently sociotechnical (not just limited to the hardware/software), the components are:

- *Need:* an identified organizational need (e.g. a business recognizes a market opportunity, or a researcher needs data processed, etc)
- *Business logic software:* a software system built around the concepts related to the business (or research) need; at this point which "tool" (whether crowdsourcing, machine learning, internal resources, or anything else) is used to meet the need is not important, as long as the need is met.

- *Workflow management software:* a software specification of the crowd workflow/algorithm (i.e. the tasks and their relationships), including quality assurance methods. This is the "control system" which turns raw crowd labor into the specific resources which meet the original need. This is now "crowdsourcing territory". There is much research on how to optimize these components [23], [53], [68]–[70], but at the same time, the person or organization with the original need typically does not have any interest in knowing all the details. As one panelist at [26] noted, "Not everyone wants to get a graduate degree in crowdsourcing just to get results from the crowd." Typical design patterns include piping results from one task to another, using gold standard tasks to train and test workers, and segmenting tasks and workers according to skills.
- *Platform:* the work platform or marketplace on which workers complete individual tasks. MTurk, Clickworker, and oDesk are examples. These provide the commodity functionality of bringing worker and task together, as well as storing data about user identity, task pricing, completion times, etc.
- *Crowd:* the crowd of workers, together. Scalability, latency or responsiveness and diversity are key features at this level of resolution.

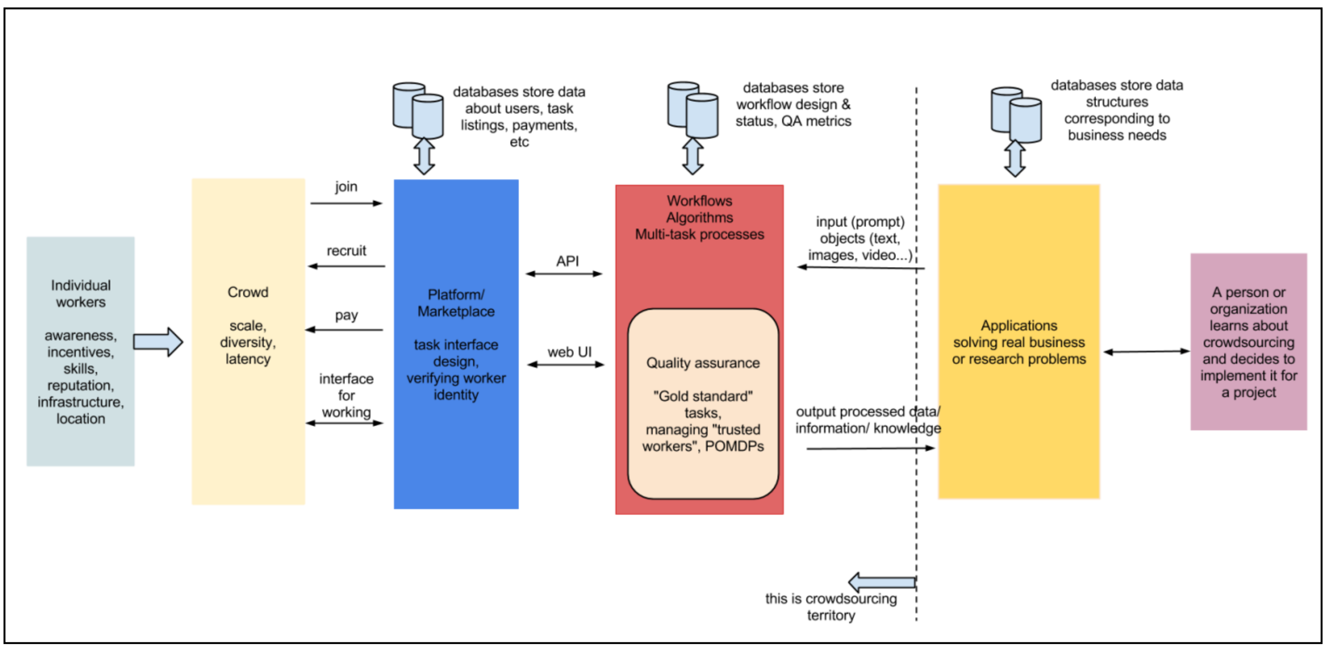



**Figure 13 The crowdsourcing stack. The request-response cycle moves from the right side to the left and back to the right.**

- *Individual:* the individual worker. To become members of the crowd and contribute to a project, workers must become aware of the opportunity, have incentive or motivation to participate, and the infrastructure to access work. Requesters are also typically interested in the skills, reputation, and location of individual workers.

Like the request-response design implemented by the web stack, the crowd stack also implements a request-response paradigm. The process goes as follows, sequentially (right to left and then left to right in Figure 13): 1) some person or organization identifies a need for human input, then 2) communicates the need and some "prompt" or "input" data to the crowd, 3) members of the crowd generate responses to the need and the data, and 4) some output data gets sent back to the requester to satisfy the original need.

Much of the challenge in building a new crowdsourcing project lies in bridging between the business need and the lower-level commodity platform where workers arrive to work (center-right of Figure 13), and this is the part of the stack that Flowbuilder addresses.

## 6.4. Flowbuilder, the crowd stack toolkit

### 6.4.1. Design Philosophy

Flowbuilder is designed to provide implementations of the common crowdsourcing paradigms for individual crowd-backed applications, for both volunteer and work-for-pay systems of incentives. It draws from both crowdsourcing research literature and practical experience to guide design choices. Flowbuilder implements the segment of the crowdsourcing stack that allows business needs to be translated into crowd workflows and implemented as individual tasks, and in the reverse direction, that aggregates individual results into responses to business needs (note that we could have framed the above as MapReduce, but computer scientists are not the primary audience that Flowbuilder targets, and our choice of terminology reflects this). Flowbuilder is crowd-agnostic; it can be connected to any existing crowd platforms (it comes with MTurk integration out of the box, and plugins for additional platforms can be added) or deployed within an organization, seamlessly joining crowd and local resources. It is not designed to be a hosted crowdsourcing platform where multiple crowd apps are co-located. Flowbuilder makes no attempt to solve problems related to task search and matching tasks to workers, though we welcome extensions to the toolkit that would address these problems. Lastly, there is a distinction among software tools between a library and a framework. A framework executes custom code provided by the developer which "plugs in" to specified locations in the framework, whereas in the case of a library, the developer's custom code executes the provided library code, allowing for more flexibility. Flowbuilder aims to provide framework-like structure to simplify the creation of straightforward crowdsourcing projects while also maintaining the library-like flexibility to allow developers to create complex, advanced crowd apps.

Flowbuilder departs from prior work on human computation workflows by deemphasizing the idea of "programming human computers" and instead accepts that humans are not computers, for better or worse. The focus is instead on soliciting high quality human judgments by adapting to users' human-ness. For example, Flowbuilder's feature set includes the ability to easily conduct training for new workers, which both improves accuracy and aligns with workers' own expressed desire to do a good job [71], [72].

With few exceptions, crowdsourcing requires members of the crowd to use a web browser to interact with a server-backed web application. Consequently the crowdsourcing stack includes as a subset the web stack, typically consisting of a server (e.g. Apache), database (e.g. MySQL, PostgreSQL), scripting language (e.g. Python, Ruby) plus a template system for generating HTML (and CSS/JavaScript). Flowbuilder builds on top of the Django web framework, which is a Python framework that includes a database API, tools for handling the http request-response cycle, and a template system.

### *6.4.2.* Features

#### 6.4.2.1. Core objects

Flowbuilder implements an abstraction of crowdsourcing tasks as follows: a task is one unit of work, the commonly referenced "HIT" (Human Intelligence Task), which can potentially be completed by multiple workers. The input data to a task is called the "prompt"; each time a worker completes a task, they submit a "result" (internally Flowbuilder avoids use of the term "input" to avoid confusion over whether it is input data to a task - a prompt - or data that has been inputted by a worker - a result). Tasks, prompts, and results are embodied via the Django object-relational mapper (ORM) as abstract base classes; this means that they provide a set of common database fields and functions in Python base classes which the developer subclasses for each type of task.

This allows the developer flexibility to create custom functionality while providing reasonable defaults for most behavior. For example, for any result, a developer will typically want to store in the database 1) the user who submitted the result, 2) the time of submission, 3) the time spent working on the task, 4) the browser's user agent, etc. This data is already handled by Flowbuilder, but the developer can add or override fields as desired. Thus at a minimum, the developer only needs to specify the types of data to be submitted by the crowd worker for each result. For each data field the developer specifies the data type with one line of code, and this automatically handles both the storage of that data in the database and the generation of the html form, eliminating the need to write and maintain code in multiple parts of the application that reference the same data field.

Each type of task also has an "interface". Flowbuilder provides a default task interface which displays the prompt data alongside the automatically generated html form for the specified result data types. The developer can also provide a custom interface for a task if more complex interactions are required. Ultimately the interface only needs to result in an http POST with form data when the user submits the task, and Flowbuilder handles data validation and insertion into the database.

Figure 14 shows how a developer defines the components of a task, and Figure 15 shows the worker's interface for completing this task. The data fields should be self-evident; for this task the developer is also providing a custom template, "instantstreetview.html", which simply specifies an iframe of instantstreetview.com with the address query defined in the template as {{task.prompt.address}}. In a similar way, the developer may reference fields on prompts and results elsewhere in the application code using python dotted-path notation, such as result.clear_view. This easy accessibility of both prompt and result data fields is what allows developers to build complex crowd workflows and easily integrate the crowd functionality into other parts of the application. The separate definition of prompts and results 1) facilitates reuse of data across the application, reducing redundancy, and 2) allows the workflow designer to conduct mini-experiments to determine, for example, which wording of a question results in better accuracy (by creating two task types with the same prompt model but different result models).

```
from django.db import models
from Flowbuilder.core.models import Task, Prompt, Result

class Address(Prompt):
   address = models.CharField(max_length=60)

class StreetviewResult(Result):
   clear_view = models.CharField(max_length=3,
                                help_text="Can you get a clear view of all"
                                       " the windows on the closest side?",
                                choices=YES_OR_NO)
   how_many = models.IntegerField(help_text="How many windows are open?")

class StreetviewTask(Task):
   prompt = models.ForeignKey(Address)
   result_model = StreetviewResult
   template_name = "instantstreetview.html"

   def on_new_result(self, result):
      …do something…
```

Generates the appropriate database field (for the chosen database backend) and the html text input widget

Generates the appropriate database field, html text input widget, and validates that the response is an integer

Defines a hook to run each time a new result is submitted for this task; can execute application code, call an external API, etc

**Figure 14 Definition of one task type, in which crowd workers would view Google Streetview imagery and answer whether or not a house has any open windows.**

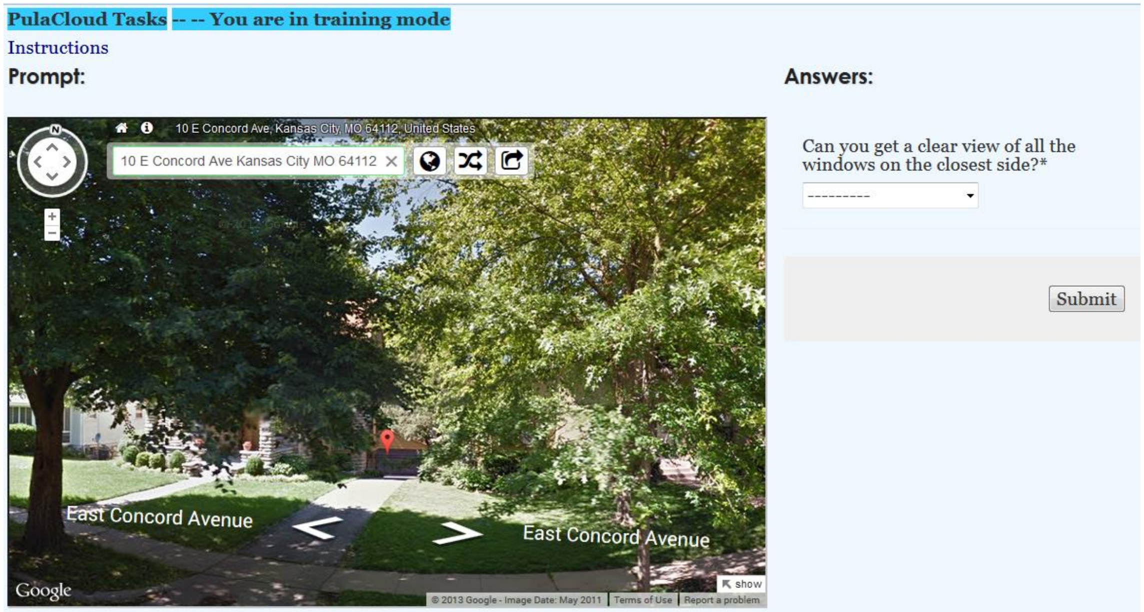



**Figure 15 Generated worker interface for Streetview Task. Note that the second question, "How many windows are open?" is hidden unless the answer to the first question is "yes". This dynamic form behavior is accomplished by specifying a Python dictionary of conditions in the Task class.**

#### 6.4.2.2. Logic

Prior work [73], [74] has shown the importance of using gold standard tasks to validate the results provided by untrusted workers. Common practice also includes showing users instructions when they first begin on a new task, and making those instructions available for later reference. Flowbuilder implements what we call the "ITT system" (Instructions, Training, Testing) for onboarding new workers which consists of first showing new workers written task Instructions (for passive learning), then proceeding to gold standard Training tasks (for interactive learning), and finally to a qualification Test. This method gives workers the opportunity to learn and improve their skills before they are evaluated. Training and learning are first-class components of the Flowbuilder system because they are essential for high-quality results on many tasks. To facilitate the creation of these gold standard tasks for training and testing, the task requester uses the same task interface as the workers in "Alchemy mode". Alternatively, the requester can import the gold standard tasks by uploading a csv file.

While the ITT system automatically handles ensuring basic competence for a particular task type, additional qualifications can be created for tasks and assigned to workers (e.g. to limit certain tasks to persons within the requesting organization). This is much the same as the qualifications implemented on MTurk, but because of the ITT system they need only be used for "special" qualifications, not for every task.

Given a task type, Flowbuilder selects a random task for a user; this behavior can be overridden to allow depth-first, breadth-first, or other strategies as desired. The task-user assignment is lightly enforced by way of being stored in the database and retrieved via cookie, to prevent users from skipping over tasks or cherry-picking the easy ones. By default a user may only complete each task once, and for each task type the workflow designer may specify the desired level of redundancy.

When a new result is submitted or a task is marked complete, the on_new_result and on_complete hooks defined on the Task class are executed. The developer can specify arbitrary actions to take on these events. Additionally, Flowbuilder comes configured with the Celery distributed task queue so that long-running tasks (e.g. calling an external API, running a machine learning classifier) can be executed outside of the http request-response cycle to prevent server timeouts. Such tasks can be initiated by the above hooks.

Flowbuilder includes a simulation module which allows the developer to simulate crowd responses to tasks to fuzz-test the workflow logic and detect problems before deploying. The module uses introspection to detect the defined field types and automatically generate technically valid (but semantically meaningless) data. Alternatively the developer can use this simulation framework in conjunction with other machine resources to submit results alongside human users, e.g. to supply machine learning classifications on a labeling task in parallel to human labelers. Together with the qualification system, tasks can be completed by human users, machine resources, or both.

Lastly, several miscellaneous but important features:

- Flowbuilder implements a simple user registration/login/profile page, but with Django's pluggable authentication architecture this can be replaced with either a custom authentication backend or perhaps eventually with an OpenID-based identity verification and reputation management system that carries information about worker skill and credibility across platforms.
- By configuring the database settings in Django, Flowbuilder's functionality can be attached to a pre-existing database, or Flowbuilder can store workflow related models in a new database while accessing business data from the legacy database.

- Flowbuilder includes the ability to upload prompt data and download result data as csv files. For continuous creation of new prompts and tasks, the developer can enable the Django REST API.
- Flowbuilder comes configured to use https for enhanced security.

### 6.5. Example use cases

In this section three real-world example uses of Flowbuilder are presented to demonstrate its capabilities.

#### 6.5.1. Finding open windows in Streetview images

An air quality researcher in an environmental engineering department wanted to know about the geographic variation in window opening behavior, in order to model in-home exposure to air pollution from indoor and outdoor sources. Using Flowbuilder, we built the task shown in Figures 14 and 15. Additionally, to collect demographic data, we created a followup task (using Flowbuilder's task piping mechanisms) for any residence for which the Streetview imagery provided a clear enough view of the home to assess all windows. In this task workers collected information about the home from its listing on Zillow.com. Implementing the Streetview window in an iframe required only overriding the "prompt" block of the task template, and for the answer form, the questions were displayed conditionally according to the specification in the StreetviewTask class (e.g. the question "How many windows are open?" is only shown if the answer to "Can you get a clear view of the home?" is yes). Additionally, the answer to the "how many" question is automatically validated to ensure it is an integer (by way of the IntegerField specification in the StreetviewResult class). For the Zillow task, no customizations to the template were required. Instructions with example images were provided (the instructions link is automatically inserted in the task page, and the requester writes the instructions in a WYSIWYG html editor on the Flowbuilder admin dashboard). Eleven training tasks were used to ensure that the workers understood the instructions and how to interpret what was meant by a "clear view" of the home. Overall, this simple workflow demonstrates the basic functionality of Flowbuilder, which allows the developer to implement a conditional two-step process with instructions and training, without having to expend effort coding each of the required components.

### 6.5.2. Responsive web design experiments

As crowdsourcing researchers ourselves, we wanted to answer the question, "How do workers behave differently if they are completing crowd tasks on a smartphone vs a laptop/desktop?" Using Flowbuilder's native capacity to adjust the task interface for large and small screens, we conducted an experiment in which we recruited 40 local workers and randomized them into phone and laptop/desktop groups, and had them complete approximately 300 tasks each of three different types. We reused the Streetview and Zillow tasks (without the conditional linkage) and added a task in which the workers rated the emotional expressions of faces. Figure 16 shows the smartphone interface for the Streetview task. For this experiment we modified the Flowbuilder system to fetch the worker's location at both beginning and end of the task to determine if users were moving while working. We encourage others to use Flowbuilder as a platform for conducting research as well;


**Figure 16 Streetview task on a narrow smartphone screen, as generated by Flowbuilder's responsive task template**

simple modifications like this can enable a wide range of experimentation while reducing the overhead of building a new crowdsourcing system from scratch for a new study.

### 6.5.3. Extracting, mapping and labeling biochemical pathway diagrams from medical literature

Bioinformatics researchers wanted to make computable the information presented in pathway diagrams in medical research articles, which are a rich source of causal biological information. The crowd app designed for this project consisted of an initial pathway mapping task in which lay crowd members re-draw the diagrams from jpg images in node-edge-node triplet format using the CytoscapeJS network visualization and analysis library. The CytoscapeJS component was provided as the task template for the mapping task (see screenshot in Figure 17). Next, these pathways were rated by expert reviewers (a native Flowbuilder task completed not by an external crowd, but by internal human resources) to identify those crowd members with above-average biology comprehension, and ultimately promote them to reviewer status themselves, as enforced by a qualification in Flowbuilder. This approach is modeled after the Shepherd system [75]. The entities extracted from the diagrams were then sent to the Unified Medical Language System's Metamap API and National Center for Biomedical Ontology Annotator (via Flowbuilder's post-result hook) which generates candidate mappings to standardized medical language ontologies. These

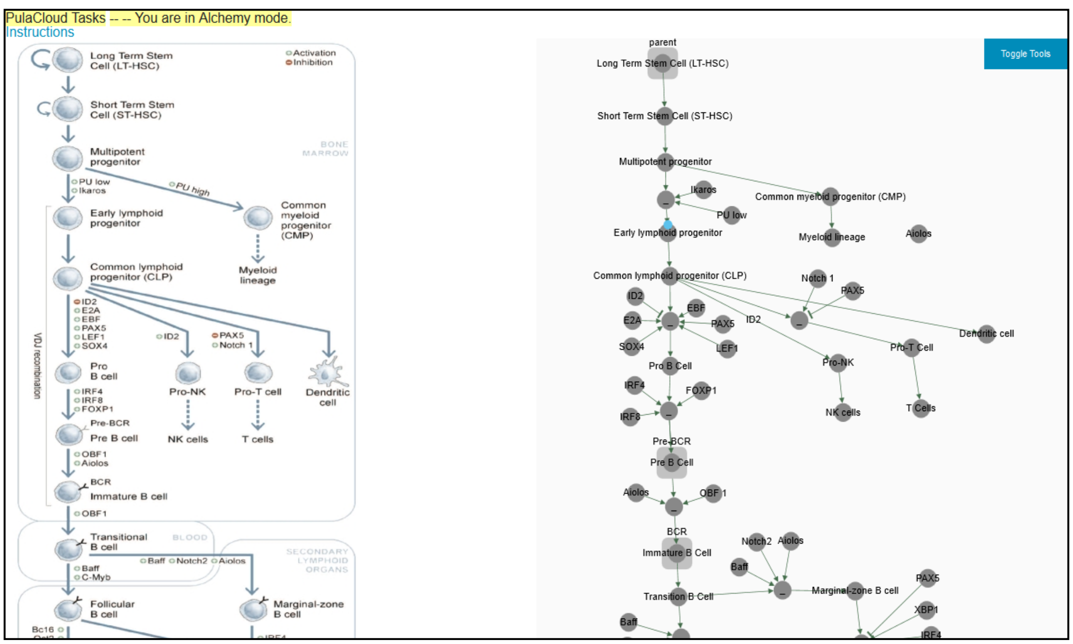



**Figure 17 CytoscapeJS interface for pathway mapping task. On the left is the jpg figure from the original journal article; on the right is the redrawn (and now computable) pathway**

candidates are then used to generate an ontology term resolution task in which workers view the context of the original journal article and figure and select the most appropriate term(s) for the extracted entity. Finally the extracted and annotated pathways are reviewed by biology domain experts before being deposited in the requesting research group's integrative bioinformatics database.

### 6.6. Limitations

Flowbuilder does not currently support synchronous collaborative tasks in which multiple workers work on the same interface simultaneously. It also does not natively support the waiting-room model for near-realtime crowd responses, though this functionality can be implemented as a custom two-task workflow. Finally, Turkomatic [76] demonstrated the possibility of workers recursively designing workflows, which is also not natively supported. We hope to address these challenges in the future and we welcome contributions from the community that implement these features.

### 6.7. Conclusions

We developed a novel conceptual model of the "crowdsourcing stack", the components a crowdsourcing application developer must synthesize to build a complete crowd-backed project. We used this conceptual model to guide the design of a developer-friendly toolkit that encapsulates the common functionality in the stack, which frees the developer from having to "get a graduate degree in crowdsourcing" to build crowd apps to solve business needs. Flowbuilder goes beyond prior frameworks for crowd workflow development by focusing on the practical concerns for getting the most value out of crowdsourcing. Developers and researchers are encouraged to use (and improve) Flowbuilder as a common resource for accelerating the development of innovative crowd-backed applications.

**This concludes the manuscripts section of this thesis.**

# 7. Future work

So far this thesis has laid out the problem of global persistent poverty, presented reasoning and some evidence for why crowdsourcing-based employment is an effective solution, and presented solutions to some of the practical sub-problems related to expanding the reach and impact of crowd work. While this approach shows promise, there is much work yet to be done to validate its effectiveness and scale it up. Future work can be divided into roughly three categories: impact evaluation, improvement of the crowdsourcing stack toolkit, and validation of the hypothesis that providing work on crowd marketplaces will incentivize potential workers to make the necessary investments to access those marketplaces (i.e. the hypothesis that targeting the labor demand side of the market will cause the labor supply side to take care of itself).

## 7.1. Impact Evaluation

I have shown for a small sample size that crowd workers use the income they receive for a variety of productive investments, particularly for education. I have not yet studied the impact of crowd work on outcome metrics related to poverty such as health status, quality of life or long term changes in household income. To do so, I recommend a cohort analysis in which the "treatment" is defined as whether or not an individual participated in a crowd work project. The same data can be analyzed as either a traditional cohort study design or using structural equation modeling to account for more complex community dynamics.

## 7.2. Development of crowd stack toolkit

Flowbuilder, the implementation of a developer's crowd stack toolkit, has been used on three example crowdsourcing projects so far. From a practical standpoint, more feedback on Flowbuilder should be collected by putting it in the hands of more crowd application developers and surveying them about their experiences developing with it. This would also give an indication of how well the conceptual model used by Flowbuilder can be adapted to a variety of real world projects, and whether the conceptual model requires further refinement in order to accurately reflect a generalized crowd work process.

### 7.3. Validation of labor demand focus hypothesis

Finally, the focus in this thesis on increasing crowd labor demand by building tools like Flowbuilder is a consequence of the hypothesis that increasing labor demand will drive increases in labor supply. This hypothesis must be validated by evaluating whether or not potential workers actually make the investments in infrastructure to gain access to crowd marketplaces once they see that there is enough work available in the marketplaces for their investment to pay off.

## 8. Discussion

Assuming that further work corroborates the effectiveness of this solution as a way to meet basic needs, scaling it up remains a challenge. While this thesis addressed some technical challenges on the path to scaling, there is much additional technical and non-technical work to be done. Because this solution is fundamentally built on economic transactions, much of this work is either work to be done by private companies or work that supports private companies. Or, put another way, *the success of CrowdWork4Dev is closely tied to the success of businesses that implement a crowd-backed approach to serving their customers.*

### 8.1. Suggestions for private companies

For companies to be successful in this space, they should specialize in providing one type of crowd-backed service (as opposed to trying to provide generic crowd services), and optimize for meeting that particular customer need exceptionally well. The market for that specialized service should be large enough to justify making the investments to optimize around it. Two examples of such markets are:

- providing business intelligence, particularly the task of finding new leads for businesses to sell to and assessing buying signals from messy data to determine which potential customers a business should target[3]
- organizing and curating biological and medical data – building knowledge models for automated reasoning systems, turning free text and other unstructured descriptions of biological processes into systems biology models, tagging entities with standard ontological concepts ("entity coreference") for search and cross-reference

[3] Author's note: the company I joined after completing this thesis addresses exactly this market.

These markets are good choices for scaling crowd-backed work because they have massive amounts of data to be potentially analyzed and the types of analysis involved often require fuzzy reasoning.

One other alternative, in an ecosystem in which many companies provide crowd-backed services for niche verticals, is for one company to provide the crowd infrastructure on which the many niche providers build. This would save the niche providers from having to deal with many issues of crowd provision and management that are not specific to their core service, much as Amazon Web Services has freed many software and technology companies from having to maintain their own physical hardware for network storage and computing. A tool like Flowbuilder provides one small piece of the overall puzzle needed to meet this need. Amazon's Mechanical Turk is an early candidate for this crowd infrastructure, and some companies (e.g. CrowdFlower, SpeakerText) have built on top of it, but it handles many crowd design patterns poorly enough that it is not a good fit for the backbone of many crowd applications.

### 8.2. Suggestions for traditional economic development actors such as NGOs and government aid bodies

Organizations that have been traditionally involved in international development work can also help drive this approach forward. To do so, they should prioritize employment-based solutions over handouts to provide beneficial rather than detrimental incentives to aid recipients. Assuming there is enough evidence to support CrowdWork4Dev as an effective solution, they could also support its further scaling by providing funding for research to help solve technical issues in crowd work at the infrastructure or application level, particularly issues pertaining to language differences and cross-cultural communication (and any other issues that arise that are specific to expanding the availability of crowd work in developing regions). Additionally, they could provide seed funding to support crowd-backed businesses that are committed to using crowd work to create employment opportunities in underserved areas. Lastly, they could contribute to raising awareness of crowd work as a positive social force, to help increase the number of users of crowd work, and eliminate the misguided concern about outsourcing and crowd work as a "digital sweatshop".

# 9. Conclusion

In this thesis I have shown that crowd work has the potential to significantly reduce global extreme poverty. Work (employment) of some kind is fundamentally required to sustainably meet basic needs, and crowd work provides the current best option for quickly distributing work opportunities to locations where 1) people are experiencing extreme persistent poverty and 2) there are not many other work opportunities.

To scale up crowd work as a solution, both the labor supply and demand sides of the marketplace must be addressed, such that they increase together. There is good reason to believe that increasing labor demand will drive increases in labor supply, though validation of this hypothesis is an important open question which this thesis has not addressed. Increasing labor demand can be achieved by 1) increasing awareness of the effectiveness of crowd data analysis and 2) putting better tools in the hands of users and potential users of crowd data analysis.

The two major contributions of this thesis are thus: 1) analysis of the interaction between the complex problem domain of poverty and the candidate solution of crowd work to determine where the most impactful place to apply solution effort would be and 2) developing a set of software tools to make implementing a crowd-backed data project faster and easier so as to increase the usage of crowd-backed analysis methods and thus increase demand for crowd labor.

## References


[1] P. Sanchez, C. Palm, J. Sachs, G. Denning, R. Flor, R. Harawa, B. Jama, T. Kiflemariam, B. Konecky, R. Kozar, E. Lelerai, A. Malik, V. Modi, P. Mutuo, A. Niang, H. Okoth, F. Place, S. E. Sachs, A. Said, D. Siriri, A. Teklehaimanot, K. Wang, J. Wangila, and C. Zamba, "The African Millennium Villages.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 104, no. 43, pp. 16775–80, Oct. 2007.

[2] J. Sachs, "The MDG decade: looking back and conditional optimism for 2015," *Lancet*, vol. 376, no. 9745, pp. 950–951, Sep. 2010.

[3] A. Basler, "The concept of integrated rural development," *Intereconomics*, vol. 14, no. 4, pp. 190–195, Jul. 1979.

[4] E. R. Carr, "The millennium village project and African development: problems and potentials," *Prog. Dev. Stud.*, vol. 8, no. 4, pp. 333–344, Oct. 2008.

[5] L. Cabral, J. Farrington, and E. Ludi, "The Millennium Villages Project – a new approach to ending rural poverty in Africa?" Overseas Development Institute (ODI).

[6] M. Clemens, "When does rigorous impact evaluation make a difference? The case of the Millennium Villages," *J. Dev.*, no. March 2012, pp. 37–41, 2011.

[7] D. McCoy, G. Kembhavi, J. Patel, and A. Luintel, *The Bill & Melinda Gates Foundation's grant-making programme for global health*, vol. 373, no. 9675. 2009, pp. 1645 – 1653.

[8] E. Duflo, A. Banerjee, R. Glennerster, and C. G. Kinnan, "The Miracle of Microfinance? Evidence from a Randomized Evaluation," May 2013.

[9] R. Hanna, E. Duflo, and M. Greenstone, "Up in Smoke:  The Influence of Household Behavior on the Long-Run Impact of Improved Cooking Stoves," May 2012.

[10] L. Beaman, E. Duflo, R. Pande, and P. Topalova, “Female leadership raises aspirations and educational attainment for girls: a policy experiment in India.,” *Science*, vol. 335, no. 6068, pp. 582–6, Mar. 2012.

[11] A. V. Banerjee, E. Duflo, R. Glennerster, and D. Kothari, “Improving immunisation coverage in rural India: clustered randomised controlled evaluation of immunisation campaigns with and without incentives.,” *BMJ*, vol. 340, no. may17_1, p. c2220, Jan. 2010.

[12] J. C. O. Esther Du O, Pascaline Dupas, Michael Kremer Ÿ, “Education, HIV and Early Fertility: Experimental Evidence from Kenya ∗,” *Am. Econ. Rev.*, 2012.

[13] J. Howe, “Crowdsourcing: A Definition,” 2006. [Online]. Available: http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html. [Accessed: 31-Jan-2014].

[14] A. A. J. Quinn and B. B. Bederson, “A Taxonomy of Distributed Human Computation,” *HumanComputer Interact. Lab Tech Rep. Univ. Maryl.*, no. HCIL-2009–23, 2009.

[15] T. Malone, R. Laubacher, and C. Dellarocas, “Harnessing Crowds: Mapping the Genome of Collective Intelligence,” *Soc. Sci. Res. Netw. Work. Pap. Ser.*, 2009.

[16] L. von Ahn, “Human computation,” 2005.

[17] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, “reCAPTCHA: Human-Based Character Recognition via Web Security Measures,” *Science*, vol. 321, no. 5895, pp. 1465–1468, Sep. 2008.

[18] Y. Yang, B. Zhu, R. Guo, L. Yang, S. Li, and N. Yu, “A comprehensive human computation framework: with application to image labeling,” in *Proceeding of the 16th ACM International conference on Multimedia - MM ’08*, 2008, pp. 479–488.

[19] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popović, and F. Players, “Predicting protein structures with a multiplayer online game,” *Nature*, vol. 466, no. 7307, pp. 756–760, Aug. 2010.

[20] “SpeakerText Inc,” “SpeakerText.” [Online]. Available: http://speakertext.com/. [Accessed: 01-Dec-2012].

[21] D. Oleson, “Crowdsourcing Scientific Research: The Crowd helping Science | The CrowdFlower Blog,” 2011. [Online]. Available: http://blog.crowdflower.com/2011/11/scientific-research/. [Accessed: 01-Dec-2012].

[22] G. Little, L. Chilton, M. Goldman, and R. Miller, “Turkit: human computation algorithms on mechanical turk,” *Proc. 23nd …*, 2010.

[23] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich, “Soylent,” in *Proceedings of the 23nd annual ACM symposium on User interface software and technology - UIST ’10*, 2010, p. 313.

[24] “Google to bankroll, build wireless networks across Africa: WSJ | Reuters.” [Online]. Available: http://www.reuters.com/article/2013/05/24/us-google-africa-idUSBRE94N0XG20130524. [Accessed: 31-May-2013].

[25] E. Brem, T. Bick, A. Schriner, and D. Oerther, “Wanted: More Nails for the Hammer — An Investigation Into the Application of Human Computation,” in *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[26] A. Bradley, E. Law, A. Kulkarni, G. Little, R. Miller, D. Weld, and B. Hartmann, “Research and Research Platforms: On Needs, Dreams, and Directions. Panel Discussion at First AAAI Conference on Human Computation and Crowdsourcing.” 2013.

[27] P. Narula, P. Gutheim, D. Rolnitzky, A. Kulkarni, and B. Hartmann, “MobileWorks: A Mobile Crowdsourcing Platform for Workers at the Bottom of the Pyramid,” in *AAAI Workshops*, 2011, pp. 121–123.

[28] A. Gupta, W. Thies, E. Cutrell, and R. Balakrishnan, “mClerk: enabling mobile crowdsourcing in developing regions,” in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI ’12*, 2012, p. 1843.

[29] N. Eagle, “txteagle: Mobile crowdsourcing,” *Int. Des. Glob. Dev.*, 2009.

[30] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton, “The future of crowd work,” in *Proceedings of the 2013 conference on Computer supported cooperative work - CSCW ’13*, 2013, p. 1301.

[31] B. Frei, “Paid Crowdsourcing: Current State and Progress toward Mainstream Business Use,” 2009.

[32] Massolution, “Enterprise Crowdsourcing: Market, Provider and Worker Trends,” 2012.

[33] A. Banerjee and E. Duflo, *Poor Economics: A Radical Rethinking of the Way to Fight Global Poverty*. PublicAffairs, 2012.

[34] W. Easterly, *The White Man’s Burden: Why the West's Efforts to Aid the Rest Have Done So Much Ill and So Little Good*. Penguin Press, 2006, p. 436.

[35] Kenya National Bureau of Statistics, “Kenya Poverty Atlas II,” 2005.

[36] T. Martindale, “Evaluation and Learning Program,” 2013.

[37] N. N. Taleb, *Antifragile: Things That Gain from Disorder*, vol. 2012. Random House, 2012, p. 519.

[38] E. Yudkowsky, “Semantic Stopsigns - Less Wrong,” 2007. [Online]. Available: http://lesswrong.com/lw/it/semantic_stopsigns/. [Accessed: 02-Mar-2014].

[39] D. W. Divelbiss, D. L. Boccelli, P. A. Succop, and D. B. Oerther, “Environmental health and household demographics impacting biosand filter maintenance and diarrhea in Guatemala: an application of structural equation modeling.,” *Environ. Sci. Technol.*, vol. 47, no. 3, pp. 1638–45, Mar. 2013.

[40] L. E. Voth-Gaeddert, A. Jobi-Taiwo, E. A. Cudney, and D. B. Oerther, "Analyzing Indicators of Multidimensional Poverty For Structural Equation Modeling Using Mahalanobis-Taguchi System," in *IWA LET2014*, 2014.

[41] M. G. Marmot, G. Rose, M. Shipley, and P. J. Hamilton, "Employment grade and coronary heart disease in British civil servants.," *J. Epidemiol. Community Health*, vol. 32, no. 4, pp. 244–9, Dec. 1978.

[42] M. G. Marmot, G. D. Smith, S. Stansfeld, C. Patel, F. North, J. Head, I. White, E. Brunner, and A. Feeney, "Health inequalities among British civil servants: the Whitehall II study.," *Lancet*, vol. 337, no. 8754, pp. 1387–93, Jul. 1991.

[43] A. Schriner, D. Oerther, and J. Uber, "PulaCloud: Using Human Computation to enable development at the bottom of the economic ladder," in *AAAI Workshops*, 2011.

[44] J. Sachs, *The End of Poverty: Economic Possibilities for Our Time*. Penguin Books, 2006, p. 416.

[45] The World Bank and World Bank, *World Development Indicators 2011*, vol. 23, no. 10. World Bank, 2011, p. 435.

[46] W. R. Easterly, *The Elusive Quest for Growth: Economists' Adventures and Misadventures in the Tropics*. The MIT Press, 2002, p. 356.

[47] S. Khanna, A. Ratan, J. Davis, and W. Thies, "Evaluating and improving the usability of Mechanical Turk for low-income workers in India," in *Proceedings of the First ACM Symposium on Computing for Development - ACM DEV '10*, 2010.

[48] F. Gino and B. Staats, "The Microwork Solution," *Harv. Bus. Rev.*, vol. 90, no. 12, 2012.

[49] P. Donmez, J. Carbonell, and J. Schneider, "Efficiently learning the accuracy of labeling sources for selective sampling," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

[50] P. Donmez, J. Carbonell, and J. Schneider, “A Probabilistic Framework to Learn from Multiple Annotators with Time-Varying Accuracy.,” in *Society for Industrial and Applied Mathematics International Conference on Data Mining*, 2010.

[51] T. P. Waterhouse, “Pay by the Bit: An Information-Theoretic Metric for Collective Human Judgment,” in *16th ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2013.

[52] V. S. Sheng, F. Provost, and P. G. Ipeirotis, “Get another label? improving data quality and data mining using multiple, noisy labelers,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, p. 614.

[53] P. Dai, Mausam, and D. D. Weld, “Artificial Intelligence for Artificial Artificial Intelligence,” *Artif. Intell.*, vol. 379, no. 8481, pp. 1153–1159, 2011.

[54] OECD, “Aid (ODA) disbursements to countries and regions.”

[55] United Nations, “United Nations Conference on Trade and Development UNCTADstat.”

[56] The Rockefeller Foundation, “Job Creation Through Building the Field of Impact Sourcing,” 2011.

[57] “Safaricom, Huawei upgrade the IDEOS to the Y100,” *CIO East Africa*, 2012. [Online]. Available: http://www.cio.co.ke/news/main-stories/safaricom,-huawei-upgrade-the-ideos-to-the-y100.

[58] W. Thies, A. L. Ratan, and J. Davis, “Paid Crowdsourcing as a Vehicle for Global Development,” in *ACM CHI 2011 Workshop on Crowdsourcing and Human Computation*, 2011, pp. 1–4.

[59] P. G. Ipeirotis, “Demographics of Mechanical Turk,” Mar. 2010.

[60] K. Land, A. Slosar, C. Lintott, D. Andreescu, S. Bamford, P. Murray, R. Nichol, M. J. Raddick, K. Schawinski, A. Szalay, D. Thomas, and J. Vandenberg, “Galaxy Zoo: the

large-scale spin statistics of spiral galaxies in the Sloan Digital Sky Survey," *Mon. Not. R. Astron. Soc.*, vol. 388, no. 4, pp. 1686–1692, Aug. 2008.

[61] "Tomnod." [Online]. Available: http://www.tomnod.com/nod/challenge/. [Accessed: 02-Jun-2014].

[62] A. Leff and J. T. Rayfield, "Web-application development using the Model/View/Controller design pattern," in *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*, 2001, pp. 118–127.

[63] A. Kittur, B. Smus, S. Khamkar, and R. E. Kraut, "CrowdForge," in *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, 2011, p. 43.

[64] G. Little, L. Chilton, M. Goldman, and R. Miller, "Turkit: human computation algorithms on mechanical turk," *Proc. 23nd …*, pp. 57–66, 2010.

[65] S. Ahmad, A. Battle, Z. Malkani, and S. Kamvar, "The jabberwocky programming environment for structured social computing," in *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, 2011, p. 53.

[66] L. Yu and J. V. Nickerson, "Cooks or cobblers?," in *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, 2011, p. 1393.

[67] A. Kittur, "Crowdsourcing, collaboration and creativity," *XRDS Crossroads, ACM Mag. Students*, vol. 17, no. 2, p. 22, Dec. 2010.

[68] M. Imran, I. Lykourentzou, and C. Castillo, "Engineering Crowdsourced Stream Processing Systems," Oct. 2013.

[69] J. Rzeszotarski and A. Kittur, "CrowdScape," in *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*, 2012, p. 55.

[70] A. Sheshadri and M. Lease, “SQUARE: A Benchmark for Research on Computing Crowd Consensus,” *First AAAI Conference on Human Computation and Crowdsourcing*. 11-Mar-2013.

[71] M. S. Silberman, L. Irani, and J. Ross, “Ethics and tactics of professional crowdwork,” *XRDS Crossroads, ACM Mag. Students*, vol. 17, no. 2, pp. 39–43, Dec. 2010.

[72] N. Kaufmann, T. Schulze, and D. Veit, *More than fun and money. Worker Motivation in Crowdsourcing – A Study on Mechanical Turk*. 2011.

[73] D. Oleson, A. Sorokin, G. Laughlin, V. Hester, J. Le, L. Biewald, and S. Francisco, “Programmatic Gold : Targeted and Scalable Quality Assurance in Crowdsourcing,” *Artif. Intell.*, pp. 43–48, 2011.

[74] J. Le and A. Edmonds, “Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution,” *... crowdsourcing search ...*, 2010.

[75] S. Dow, A. Kulkarni, B. Bunge, T. Nguyen, S. Klemmer, and B. Hartmann, “Shepherding the crowd,” in *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA ’11*, 2011, p. 1669.

[76] A. P. Kulkarni, M. Can, and B. Hartmann, “Turkomatic,” in *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA ’11*, 2011, p. 2053.

[77] D. Schaafsma, R. Gosens, I. S. T. Bos, H. Meurs, J. Zaagsma, and S. A. Nelemans, “Role of contractile prostaglandins and Rho-kinase in growth factor-induced airway smooth muscle contraction.,” *Respir. Res.*, vol. 6, p. 85, Jan. 2005.